

SINCLAIR

2 4 6 7-10 13 14
18 20 23 32 33 36
38 40 45

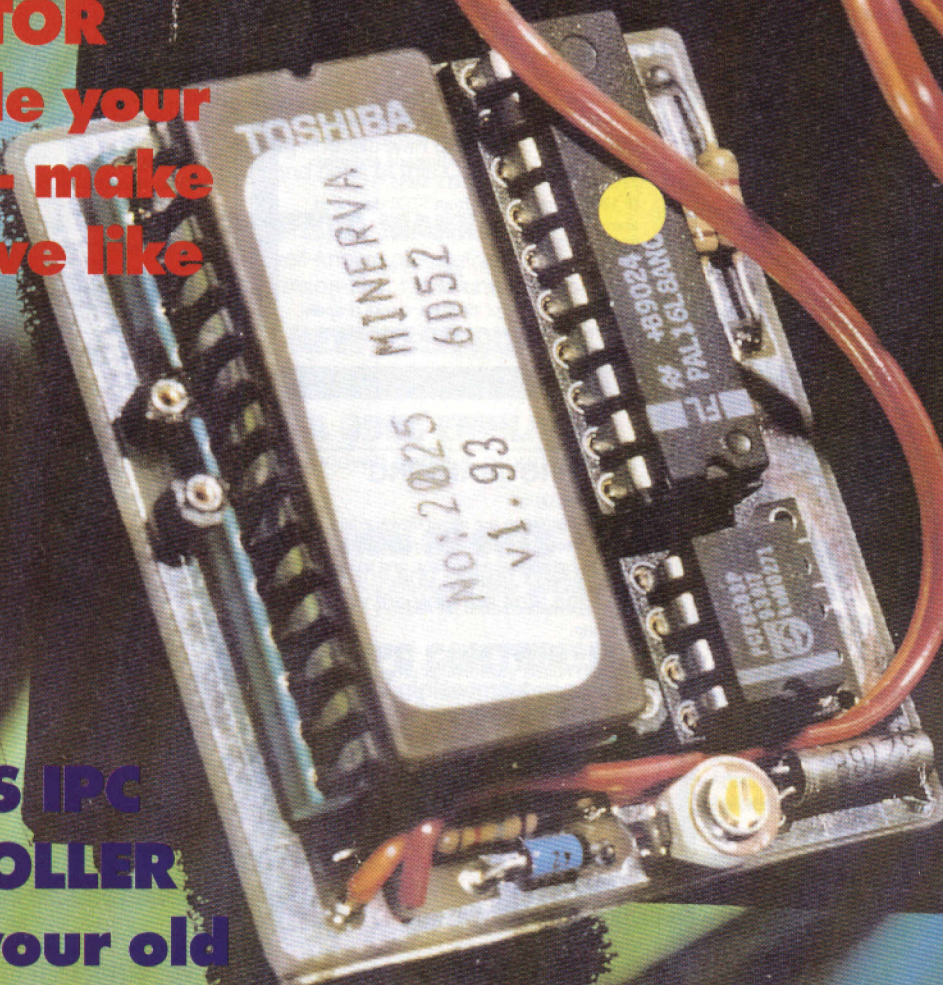
WORLD

OCTOBER 1992 £1.95

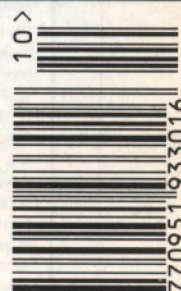
MINERVA ROM MK II
This is not The One
- This is The Two!

**AMIGA QDOS
EMULATOR**
**Upgrade your
Amiga - make
it behave like
a QL**

**HERMES IPC
CONTROLLER**
**Make your old
8049 nervous**



ISSN 0951-9335



9 770951 933016

Editor
Helen Armstrong

Publisher
Mark Kasprovicz

Advertising Manager
Jim Peskett

Creative Director
John Stanley

Designer
Steve Billington

Magazine Services
Pauline Wakeling
Frances Maxwell
Linda Miller

Sinclair QL World,
Published By Arcwind Ltd.
The Blue Barn,
Wootton, Woodstock,
Oxon. OX7 1HA
Tel: 0993 811181
Fax: 0993 811481
ISSN 026806X

If you have any comments or difficulties please write to the Editor and we will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.
Back issues are available from the publisher price £2.50 UK, £2.99 Europe. Overseas rates on request.

Subscriptions from: Arcwind
The Blue Barn, Wootton, Woodstock,
Oxon OX7 1HA
UK: £23.40
Europe: £32.90
Rest of World: £40.90

Reprographic Services: The Wace Group, London.

Printed and Bound by
William Gibbons, Planetary Road,
Willenhall, West Midlands

Distributed by: Seymour Press Ltd.,
Windsor House, 1270 London Rd,
Norbury, London, SW16 4DH

© 1992 Arcwind Ltd,
Sinclair QL world is published 12 times a year. All rights reserved. Reproduction in whole or in part without written permission is strictly prohibited. We welcome contributions but can not be responsible for any unsolicited contributions, all material must be supplied with a SAE.

CONTENTS

■ ■ OCTOBER 1992

4 TROUBLESHOOTER

Please send your address with orders and enquiries! And Bryan Davies finally confronts the dreaded Serial Link.

11 QL SCENE

Miracle plan three new products for end of 1982 ... Quanta meeting in Thetford ... All Formats Fair dates ... Printmaster from DJC

12 OPEN CHANNEL

Disks fixed ... Manual ... Archive ... Xchange Group gone ... descending in Abacus.

14 PRINTMASTER

A new program from DJC allows you to set your printer up from a menu. Henry Orlowski puts it through its paces.

16 3D TERRAIN - UK

Making shapes with data input is a popular pastime, it seems. Bryan Davies tries the pros and cons of this CGH release.

20 MINERVA MARK II

The well-known alternative QL rom has moved on from its many versions-one to its first version-two. Ian Bruntlett runs it.

23 THE NEW USER GUIDE - part 19

Section Nineteen follows on directly from last month's instalment with FORMAT #chan to HEX\$.

28 AMIGA QDOS EMULATOR

Simon Goodwin runs QL programs on the Commodore Amiga with the latest emulator software.

32 A QUESTION OF DOTS - part 3

Beginning at the end - Geoff Wicks explains how page design in desktop publishing is determined by the way the final copy will be reproduced.

35 HERMES IPC CONTROLLER

The Hermes is a replacement upgrade for the QL's 8049 Intelligent Peripheral Controller chip. Ian Bruntlett tries it for size.

36 FONT EDITOR

Rich Mellor introduces a public domain routine to help you customise your fonts.

38 PSION SOLUTIONS

Howard Clase presents some routines to make life with Psion even psunnier; including OPENing files straight from the medium to view them.

40 PERFECT PARAGRAPHS AND SHINING CHARACTERS

Driving a printer is not enough for Martin Harris. He teaches his printer perfect deportment as well.

COMING SOON!

Mike Lloyd is still pursuing the

SER Mouse, though not, we hope, with a carving knife.

TROUBLESHOOTER

How often have you bemoaned the fact that the Psion programs were never updated after the first year or so? By the time we discovered what was needed, Psion had gone on to other activities and didn't want to know. We have now got good WP programs to replace *Quill*, but we never have had a complete replacement for *Archive*, although *FlashBack* does a very good job of some of *Archive*'s functions. PC users who still support Psion have been favoured with an updated *Archive*. In principle QL *Archive* Procedures can be transferred to the PC version using a transfer routine such as *XOver* (supplied with DP's PC emulators), although some of the PC *Archive* command language differs from the QL version. Does anyone have experience of using the updated PC *Archive*, and of transferring Procedures from the QL to it?

Ready to go

As of the third week in August, Digital Precision say they have shipped any outstanding orders for Special Edition versions of various programs, with the last-minute updates completed. The one exception is *QMaths Two*, which is getting further attention now the others have been cleared out of the way; this is a program which requires plenty of quiet! Apart from having "Special Edition" tagged on the end, the programs have the same names, but that doesn't mean they are the same as before; there have been major changes to such as *Conqueror* and *Transfer Utility*, adding considerably to their usefulness. Detail changes include revision of parts of the *Perfection* instructions, with a 7-page index being a welcome addition. A further 15 pages deal with the differences between *Perfection SE* and the basic version. For those who follow the rapidly-changing version numbers of WP programs, *Perfection SE* has reached 4.12 (at least!)

Clarification

Following my review of *text87 Plus4* in the August *Q L World*, Software87 called to discuss the mechanism for printing two A5 pages on one landscape A4 sheet of paper, leaving enough clear space down the middle for hole-punching. The problem as I found it at the time was that one can't specify five columns, and four columns must have equal spaces

Bryan Davies gets himself deeply involved with Serial links.

CONNECTIONS FOR SERIAL QL-TO-PRINTER CABLE

Note that this cable was made up for an Epson GQ-5000 laser printer and other printers may (but shouldn't) have different pin assignments. The connections shown apply to both the PCC ("telephone") connectors used on U.K. QLs and the 9-pin D connectors used on U.S. & German QLs. The only reason for showing pins 1 & 7 connected together on the 9-pin connector is that a Miracle Systems serial-parallel interface for the U.S./German QL has this connection, but the equivalent connection is very likely made internally on those QLs anyway. The connections shown are for the SER1 port on the QL.

QL-END CONNECTOR	QL SIGNAL	PRINTER-END CONNECTOR
6-PIN PCC MALE or 9-PIN D MALE		25-PIN D MALE
1	GROUND	7
7 (not PCC)		
3	RxD/OUTPUT	3
4	DTR/READY INPUT	20

NULL-MODEM SERIAL PORT LINKING CABLE

The U.S.- & German-market QLs have 9-pin female D serial port connectors. PCs commonly have either (or both) 9-pin male D or 25-pin male D connectors. Whether or not the pin connections for QL & PC are the same is an open question; likewise with the connections to the 6-pin PCC connector used on U.K.-market QLs. The cable connections shown below are for PCs.

25-PIN D FEMALE to 25-PIN D FEMALE	SIGNAL
7	GROUND
2	RECEIVE DATA
3	TRANSMIT DATA
4	CLEAR TO SEND
5	REQUEST TO SEND
6	DATA TERMINAL READY
20	DATA SET READY

NULL-MODEM SERIAL PORT LINKING CABLE

The connections shown below are for PCs. To connect a QL to a PC, use a male 9-pin D (U.S. & German versions) instead of the female, or a male 6-pin PCC (U.K. version). There may be some differences in pin-assignments on the connector for the QL. In particular, the QL uses 1 (PCC) or 6-7-8 (D) for ground. See other illustration for comparison between D & PCC connections on the QL.

9-PIN D FEMALE to 25-PIN D FEMALE	SIGNAL
5	GROUND
3	RECEIVE DATA
2	TRANSMIT DATA
7	CLEAR TO SEND
8	REQUEST TO SEND
6	DATA TERMINAL READY
4	DATA SET READY

NULL-MODEM SERIAL PORT LINKING CABLE

This cable is for PCs not QLs. See comments in other illustrations for differences in QL 9-pin connections.

9-PIN D FEMALE to 9-PIN D FEMALE	SIGNAL
5	GROUND
3	RECEIVE DATA
2	TRANSMIT DATA
7	CLEAR TO SEND
8	REQUEST TO SEND
6	DATA TERMINAL READY
4	DATA SET READY

between them, resulting in either an excessive gap between text columns or too-small a gap in the middle of the A4 sheet.

As was pointed out in the example given, you can - at the expense of taking more time - use Frames instead of Columns, and this removes the restriction on the number of vertical areas on the sheet. You could specify one frame with two columns in it, for the first A5 page, then a frame with one column and no text in it for the hole-punching space in the middle, then a further frame with two columns for the second A5 page. In the example in the review, the total number of frames required would be seven (four for headers and footers), which is quite acceptable within text87 Plus4.

What do you expect?

A couple of letters have been passed to me by Digital Precision, with the comment "what can you do with letters like this?" The letters are dated exactly a year apart, and both suffer from the same "defect" - they don't have the writer's address on them. The first didn't have a signature either! The name on the second letter was not known to DP or to me, and the writer resides in another EEC country, making it rather difficult to track down the address. Fortunately, some sleuthing work by DP has yielded an address and the writer will be contacted about the problem he wrote of (connected with printing on a 24-pin Epson LQ-500 from *Professional Publisher*). He will also be advised to be careful and buy his programs direct in future, as DP has no dealer in the country concerned; you cannot realistically expect support for a program when it hasn't been bought from a reputable source.

Another letter, from a different country, suggested a lack of forethought on the part of the writer, as he apparently expected a QL program to be supplied on IBM-format disks. While we have heard much more of PCs in the QL world since the advent of PC-

emulation programs for the QL, there is a considerable difference between program files for the two computers, and you can't transfer them directly from one disk format to the other.

Faced at last!

It may seem a bit ridiculous, but the thought of making a printer connection by a Serial link has always caused me to look for another job to do. Whereas using a Parallel connection involves nothing more than plugging the cable in, the volume of instructions on the setting of parameters, which is provided with printers and computers for the serial connection, is enough to put off even hardened hackers. As Miracle offered a serial-to-parallel interface from the early QL days, there has been little reason to bother about the fact that the printer output on the QL is from a serial port, rather than a parallel port as is usual on many other computers.

Having, therefore, avoided the problem for nearly eight years, I finally faced up to it when yet another device took up residence on my workbench and gave me the task of connecting three computers to one printer. It could have been done with the two existing 2-way switch boxes, but that would have meant re-siting one of them, and space is short. The printer has both serial and parallel interfaces built in, so why not make use of both?

The hardest part was discovering what the wiring connections had to be, for the connecting cable. The QL in question is a US JS version, with 9-pin D female connectors for the ports, and the printer has the standard 25-pin D female connector. My various notes, made over the years, were not entirely in agreement with each other, but the arrangement shown in the illustration worked fine. Don't be fooled into thinking that a manufactured cable with 9-pin male on one end and 25-pin male on the other will automatically have the correct connections for this purpose; the cable I used was one purchased at a computer

fair, and may have been made for a Spectrum, but it was wired quite incorrectly for use with either QL or printer.

Having plugged the cable in, the first test was a simple one: to establish whether or not there was any communication between QL and printer. This was done by typing the following on the SuperBasic command line:

```
OPEN#7,SER1:PRINT#7,"Ar  
e you hearing me?":CLOSE#7
```

Rather to my surprise, the text between the quotes was printed, in Courier 10-cpi font (the default setting).

Document

Next step was to load a one-page document into text87 and try a print from that. After a false start because of failure to use the CLOSE command following the first test, the document was printed with no hassle. Further tests were made using the SDP (screen dump) command in the Gold Card, with no problem. It really was that simple - no setting of XON/XOFF, parity, stop bits, etc., at either end of the link. Both ends subscribe to the same philosophy and seem to get on well together. The only hitch now comes when I forget to change the printer switch setting from parallel (the default) to serial - you can waste a lot of paper that way.

Next step is to investigate transfer of files via the serial ports, from QL to PC and vice versa. For this, you need a "null-modem" cable. There is a gender difference here, as the PC serial port has a male connector, whereas the QL uses a female. The connecting cable needs a 9-pin male D or 6-pin male PCC (depending on national version) for the QL end and a 25-pin female for the PC end. The only connection in common with the printer cable is between the pin 7s, which are the ground link; there is no pin 7 on the PCC connector, though. Not surprisingly, for two computers to talk to each other through the same ports, it is necessary for the signal connections to be reversed between the ends; what is connected to input at one end

has to go to output at the other. Unfortunately, the various documents in my possession do not use common terminology for the signals. The QL User Guide hints darkly that others interpret the RS-232 specification incorrectly but, with the number of PCs in the world now quoted at 100 million, we would be fighting a bit of a battle to persuade everyone else to adopt the QL standard. We are stuck with the problem of trying to adapt wiring diagrams taken from other computers.

From Conqueror

The illustrations given here are the same as recent purchasers of Conqueror Special Edition will find in their DR-DOS 6.0 reference manual, in the section on the FILELINK command. That is, they are strictly for PCs. The 25-to-25-pin link connections have been tested on my PCs and seem to work. It is definitely necessary to give some thought to the 9-pin connections, as the Sinclair view of a 9-pin serial port differs from the standard IBM one. RS-232 is not always RS-232! Please use the illustrations given here only as a starting point - they all (except one) refer to PC connections.

The odd illustration is the one which relates the two SER1 port types on the QL to each other - PCC to D. Even here, there is some apparent inconsistency, as 1 on the PCC is ground, whereas 1-6-7-8 are ground on the D, and 6 on the PCC relates to 9 on the D. In addition, the signal designations given in the *User Guide* don't match fully with those in, for instance, the DR-DOS guide. Any readers who have information that they can vouch for, on these connections, please write in and give details.

What has been tested on my systems, and works in a limited fashion, is the link between two PCs, using a 25-pin female-to-female cable. As with the printer link, everything went well at first, so it wasn't any surprise to then discover an annoying problem. Referring back to the DR-DOS

6.0 operating system that comes with Conqueror, one of its major features is a file-compression utility which allows an effective increase of more than 50% in available disk space. Unfortunately, the second PC in my test wasn't

using DR-DOS and didn't want to know about compressed drives. After verifying that there were the correct number of directories on those drives, the software in use threw up its hands in horror and said goodbye.

Speed increases

The theoretical advantage of parallel transfer over serial is that of 8 bits at a time against 1 bit, and the lesser overhead of the transfer protocol. In practice. The devices

connected to the ports have been slow enough to make the difference between the methods of transfer relatively unimportant, until fairly recently. With the increase in the use of modems, and remote connection by line in general, there has been great pressure to increase the data transfer rate through serial ports and the inter-connecting lines. Referring again to the DR-DOS book, the maximum possible rate given for file transfer using this

function is 115,200 baud, which presumably equates to 11,520 bytes per second. There was no apparent problem transferring files at this rate with the serial cable between my systems. The top rate quoted for the QL serial ports is 19,200 baud, or 1,920 bytes per second, but you won't normally get above 9,600 baud.

Obviously, such high rates are useless for devices like printers which can't cope with them, but modems have got decidedly speedy recently and file transfer needn't make British Telecom quite so rich.

Miracle Systems obviously know there is still a market for their serial-parallel interface. This is the standard method of connecting QL to printer. The price and package have remained remarkably constant over the years, but the package has been considerably miniaturised recently, to the extent that the electronics have become part of the parallel connector - just about as convenient as you could get.

COMPARISON BETWEEN QL PCC & D SERIAL PORT CONNECTIONS

These pin-assignments are for the female connectors built-onto the QL motherboard, as checked on sample JM (U.K.) & JS (U.S./German) machines. The signal designations are as printed in the QL User Guide, Concepts section, Communications RS-232-C sub-section. The terminology is different from that used in typical PC books, the connectors are female whereas those on PCs are male, and some of the pin-assignments differ between QL & PC. The connections shown here are for SER1.

6-PIN PCC FEMALE	NAME/SIGNAL	9-PIN D FEMALE
1	GND/signal ground	1
		6
		7
		8
2	TxD/input	2
3	RxD/output	3
4	DTR/ready input	4
5	CTS/ready output	5
6	-/+12V	9

plus4

A Quantum leap in QL Wordprocessing

Users have expressed their verdict in letters published in Quanta!

+1 USER FRIENDLY TO THE EXTREME

Context-sensitive help * File selector boxes * Highly compatible with Quill.

+2 THE MOST POWERFUL QL WP

Reformats as you edit * Remembers paragraph formats between sessions * Integrated spell checker with dictionary browse and replace * Multiple Windows over multiple documents allow you to edit one text while looking at a different one for reference * On-screen page-preview and pagination accounting for columns and variable line-spacing.

+3 UNRIVALLED PRINT QUALITY

Nothing else can compete in text and character formatting * Fully supports proportional spacing, mixed character widths and justifies correctly * Multiple paragraph formats with different margins and line-spacing for each * Right, centre and decimal tabs * Multiple columns plus headers and footers * Desktop publishing with several different page layouts in the same document. All the above features available at the highest text-mode letter quality resolutions of 360 x 360 dots per inch (24-pin printers) or 300 x 300 dpi (deskjet and laser) at fast print speed and with very large texts.

+4 FASTEST QL WORDPROCESSOR

Amazing timings on QL and 8 MHz Atari ST with QL emulator. Gold card would be much faster than both * A 70 page text (24,000 words, 141,000 characters) was used * Load (first time from floppy disk--not ram disk or slave blocks): 25s (STQL 17s) * Save (to floppy disk): 37s (STQL 32s) * Search / Replace 580 instances (auto reformats the affected paragraphs to account for different lengths of search and replace strings--not leaving

you with hundreds of lines with ruined format to sort out manually): 43s (STQL 14s) * Change right margin of whole text (with complete reformat): 65s (STQL 17s) * Move 10 pages from top to bottom (times include manual marking and positioning of the text and reformat after the move): 35s (STQL 15s) * Scroll full screen over text 100 lines (full screen line by line--not just the last line, not just the last line over every 10th or 100th line of text): 19s (STQL 5s) * Change justification of whole text (change is instant, times represent full screen refresh after the change): less than 2s (STQL 1s) * Many other operations (e.g. goto page, line, top, etc., case sensitive search, insert typefaces, auto reformat of edited paragraph, change individual paragraph margins) are instant, with only a quick screen refresh to complete the task.

Fully compatible with all QL ROMS, Gold Card, STQL (also STE and high resolution), Thor. Requires disk drive and 256 K memory.

Prices payable by cheque (Air Mail overseas)

text87plus4	£79
2488 drivers for 24-pin and bubblejet printers	£19
typeset90-deskjet drivers for all HP Deskjets	£19
typeset90-Epson drivers for Epson Lasers	£39
fountext88 + founted89 graphic driver	£39

Also: QDesign II: £53 * QSpread: £49 * QPac II: £39 * QTop: £29 * QD4: £39 * ArcEd: £19 * FiFi: £19 * Mouse £49

Software87, 33 Savernake Road, London NW3 2JU

DEMO!

We have produced a working demo version of plus4 to try for yourself. The demo has all the commands apart from save, export, spell check and print (sample print output for different printers is included). Send £2 in postage stamps (no cheques please) to get your disk and leaflets.

Digital stock up

Digital Precision has been in contact to tell us that, for direct transfer customers, their bank details have undergone some changes. The business's Sort Code is now 20-53-00, and their account number is 60327808. All other details remain the same - the alterations were brought about by their branch's change of status to a Business Centre instead of an ordinary local branch. Only the numbers have changed.

Digital Precision have also managed to buy up some old warehoused stock of software previously published (or planned for publication) by Pyramide, Eidersoft and Compware. The software is a "very mixed lot which needs to be sorted out". *QL World* understands that DP will be advertising the stock when they have decided what is suitable to be put on the market.

Miracles never cease

Miracle Systems are, it seems, having a busy Autumn. They were contacted earlier in the year by an engineering student from the Netherlands, who is also a QL enthusiast. The result is that Noud Snelder, from Arnhem, is spending six months working with Miracle on the development of their current projects, as part of his engineering sandwich course.

Miracle hope to get three important new products for the QL ready by the end of the year, although, says proprietor and prime mover Stewart Honeyball, "we can't guarantee that". However, to keep the ball firmly rolling in production of their best-selling Gold Card, and development of the new repertoire, Miracle are also retaining the services of a full-time office assistant, and two outworkers assembling Gold Card hardware for the company, for the next few weeks. "The Gold Card has been selling a bit quicker than we expected," opines Stewart, who pursues a firm policy of warning customers if there is a waiting list for products.

The projects in development are a SCSI interface, aimed mainly at hard disk users, but also for floppy disk drives; a plug-in card to enable PC users to run QL software directly on their PCs, using the PC's own 3.5in disk drives and other peripherals; and the long-awaited graphics card for the QL.

Stewart notes that the PC/QL board plugs into a standard expansion slot in the

PC, to enable users of both computers to run QL software on the PC's hardware; it is not an emulator; and it does not utilise the QL's own hardware.

Miracle will be showing their current products at the **Quanta workshop** in Thetford on 3 October; the Scottish All Formats Fairs in Edinburgh on 10 October and Glasgow on 11 October, and in London at the Novotel, Hammersmith, on 17 October. Miracle is making an effort to co-ordinate visits of QL traders to non-QL-specialist shows like the All Formats to ensure that QL users know where they can meet their suppliers.

With this much on their hands, it's no wonder that Miracle have licensed out some of their older best-sellers, like the Trump Card, to Qubbesoft to market and support.

Master your Printer

"Is your printer dominating you, when you should be in control of your printer?" asks Dilwyn Jones in his introduction to DJC's latest utility, *Printmaster*.

Print master lets you set the functions on your printer by selecting the ones you want from a menu, and pressing Enter - far easier than finding and typing in codes from the manual.

The drivers supplied with the program will work with most printers, or you can create or adapt your own driver from within the program. The supplied drivers cover printer from quite early, simple ones, through many 9-pin printers to some 24-pin printers. They can be easily adapted.

You can set up combinations of functions, and a carefully-written manual helps to ensure that the program is easy to use, even for the less experienced printer user.

Printmaster is reviewed this month on page 14. It costs £20, is available on 3.5 or 5.25in disk or microdrive, and can be used on an unexpanded QL. Orders and enquiries to **Dilwyn Jones Computing, 41 Bro Emrys, Tal-y-Bont, Bangor, Gwynedd LL57 3YT. Tel. 0248 354023.**

QL SCREEN

All Formats Computer Fair Dates

Talking of the All Formats Computer Fairs, the latest schedule of events we have is: October 10, Assembling Rooms, George St., Edinburgh; October 11, Candleriggs, Glasgow; 17 October, Novotel, Hammersmith, London; October 18, Brunel Centre, Templemeads Station, Bristol; 24 October, Haydock Park Racecourse, J23 of the M6; November 1, University Sports Centre, Leeds; November 6, National Motorcycle Museum, J6 of the M42; November 7, Sandown Park Racecourse, Esher, Surrey J9/10 M25 (or train from central London); December 5, National Motorcycle Museum.

The Fairs run from 10am to 4pm. Tickets are £4. Stands and other information from **Bruce Everiss on 0608 663820.**

QL traders are now tending to concentrate on the London and Scotland fairs.

Quanta in Norfolk and Avon

For information about the **Quanta** meeting in Thetford, Norfolk, scheduled for October 3, contact **Geraint Jones on 0842 762406.**

Another Quanta meeting is planned by the Bristol Quanta group on Sunday 29 November, from 10am, at the Walton Park Hotel, Clevedon, Avon, south of Bristol on the M5 motorway.

The Group will be putting on their famous Bring and Buy sale, large-screen presentations, demonstrations and traders. Further information from **Mike Ashford on 0272 629981.**

OPEN CHANNEL

Open channel is where you have the opportunity to voice your opinions in Sinclair QL World. Whether you want to ask for help with a technical problem, provide somebody with an answer, or just sound off about something which bothers you, write to:

Open Channel, The Blue Barn, Tew Lane, Wotton, Woodstock, OX7 1HA.

Den'll fix it

I wrote to QL World some time ago about my disk drives and my cat. You published the letter a couple of months ago. I have now managed to get the problem fixed, within 24 hours, after the last two computer repairers I contacted said it couldn't be done, or wasn't worthwhile. Who do I have to thank? Someone just down the road from me - Mr Briggs of Adman Services. I've had no further problems with the drives - or the cat - since. I had previously been without a disk drive for around six months and had bought a replacement unit.

I would also like to say how glad I am that Sinclair QL World has survived Mr Maxwell. As a computer science graduate I still find much of interest between the covers of QL World, but find the product reviews most helpful. I have use of a variety of different computers, and apart from the poor graphics, still prefer to QL to any of them.

**M J Williams
Telford
Shropshire**

Comment: Now don't say we never mention Adman Services ... Dennis related some of the sorrows of a disk-drive-doctor in a recent Quanta bulletin, from which one can deduce some good principles, of the following sort:

(a) Don't buy dodgy-looking gear and expect it to work properly (b) find out if

your drive and interface work (especially together) before you buy them - don't just assume that because you've heard that both items worked somehow, somewhere, that they do now (c) read the instructions (or listen to your supplier) (d) your friendly expert will try and help but can't necessarily fix everything, especially in his spare time and (e) don't spray anything at random into delicate machinery. (Even water, may I add.) I'll add one more of my own: if something doesn't work, its better to pay someone to fix it than it is to poke about in its innards. Get a quote first; try to get it done through recommendation (or do what MJ did, and talk to Adman, TF Services or someone else you've heard of); and if you really don't know its history and why it doesn't work, recognise that you may be in for a disappointment.

But what did Dennis do to the cat?

Manual

I find the old Serial 8056 printer very useful for printed output of listings and so on, but have found it irksome to change the plugs inserted in Ser1 for this printer and for my other printer, a Mannesman MT81. The lead from the 8056 printer has a 16-pin plug and socket, and all the necessary changes to the connections are made within the body of this 16-pin plug, which is connected to the six-contact plug going normally into Ser1.

Reference to page 13 of the Concepts section of the *QL User Guide* shows that for Ser2 to act as an output port, the leads from pins 4 and 5 have to be interchanged (green and yellow wires); secondly, output has to come from pin 2 instead of pin 3. This is done simply by unsoldering the red lead, bending it back out of the way and joining a short length of wire to the black led, which has been cut off short, and connecting this to where the red lead was formerly.

I now have the 8056 plugged into Ser2 and the MT81 into Ser1; I have versions of the Psion Quartet

for each printer, and using either these or directly keyed commands, the above modification is proving a godsend. My thanks to David Johnson of Quanta for reminding me (to my shame) that the Manual does exist.

**JHS Neilson
Kettering
Northants**

Editor's comment: The most popular questions crop up over and over again - not all of them are resolvable by referring to the User's Guide, but it is a good place to start.

PC Archive

I use the PC-Four version of Archive regularly at work, while using the QL version at home. I introduced Archive to my boss (a dBASE fan), who was impressed by its versatility and the effects I made with its screen driver. He went ahead and bought PC-Four, and I'm at present programming some important applications using it.

This made me investigate some ways to exchange information between my QL and the PC I use at work. I've been able not only to exchange a database, but a program with all its procedures to make corrections at home, and then re-load them in the PC at work. I will describe now how I do it.

To transfer a database from QL to PC, you must first EXPORT that database to a floppy disk. This will create a file in import/export format (refer to the *QL User Guide*), with all the data. This is the file you must transfer to a disk in PC format. Then load Archive in a PC and IMPORT that same file. You will have the same file in your PC and your QL.

The index system works differently in the two versions of Archive: while in your QL, you have the index scrambled with the data in one file; on the PC, you can have up to five indexes, and choose to use any of them. They reside in separate files with extensions IX1 to IX5. This is the data you cannot transfer between computers. You must re-index using the command ORDER in

the computer, into which you have IMPORTed the file.

Something similar applies to programs. I list them to disk using SPOOLCON and transfer it to PC format after converting some codes such as end-of-line, and others with Xover. Then in the PC I just LOAD this file.

Joaquin Herrero Pintado
Madrid
Spain

Group gone

You give some addresses towards the back of *QL World*, which I have found of considerable use. Now, may I draw your attention to one that didn't work? The Xchange Users Group. I enclose the Post Office's normal enlightenment - a returned envelope with a bright orange 'gone away' sticker on it.

I tried many times to get Psion themselves by phone, to find out what the Psion Four for an MS-DOS machine would cost, going through a number of apparent changes. Finally, I got them on 071 262 5580. They told me that I would get the suite more cheaply directly from a seller, and that I should try Frontline on 0256 463344. The people on this number said that they are only distributors - I should try 'Solicitors' Own Software' on 0225 448664. Bingo! Cost of the Psion suite? £495. I have a considerable amount of reference work in *Quill*, and some of my customers want to use MS-DOS, but £495 would be rather a lot for me to add to the price of a job! Aren't we lucky to get the QL SuperBasic versions so reasonably!

Robert Taylor
Swansea

Tables turn

Readers who have looked at my article on using Abacus for keeping a check on a bank account may have been puzzled by the absence of Tables 4 and 5 in the September issue.

The article was originally submitted in two versions - as a single article, and in two parts, with Tables 1 to 3 for the latter and Tables 1 to 5 for the

latter. In the second version, Table 2 was split into three parts, with quarterly and bi-annual formulae extracted from Table 2 (as published) forming Table 4. Table 3 thus became Table 5.

Obviously, the trauma of the magazine's move from one home to another was responsible for the tables for the single article being printed with the two-part version. However, I can assure readers that the aggregate contents of the Tables are identical, and I hope no-one has been put off by the mishap.

P H Warne
Newcastle Upon Tyne

Comment: I think, in fact, that this was a trauma caused by having more than one version of the article! Not all variant versions arise from something as helpful as having a choice of a one- or two-parter - they usually arise from authors', ahem, updates. The big tables needed a certain amount of cut-and-paste to get the originals to a reasonable size, and I eventually decided to run the most readable pair in both installments of the article - but obviously forgot to change the cross-references. All the information should be there.

sucabA

Your reader, E E Stocker, asks how to arrange *Abacus* entries into descending order rather than ascending (*QL World* July 1992). Here is one way of doing it:

Select a column remote from the main spreadsheet - let's say column Z - and arrange for the negative value of the entry in the column to be ordered (say column AA) to be entered in column Z.

All that is required is a formula ($A1 * -1$) to be in cell Z1 and for it to be ECHOed down column at least as far as the last full cell in column A.

When that is done, press F3 and type ORDER, Z from A1 to A10, and column A will be put into descending order.

C B Storey
Tyne & Wear

Reference E E Stocker's letter regarding the use of Abacus:

An easy way to produce ORDERing in reverse sequence is to:

- 1) Order the desired column on ascending value
- 2) Add a temporary column at any position
- 3) Enter sequential numbers preceded by a minus sign into the temporary column (ie col=-row() from 1 to 20)
- 4) Order the temporary column, and then delete it.

The problem of Abacus failing to PRINT formulas over 60 characters long can be overcome by the following:

1) Enter each cell which contains a formula and amend the formula to a string by adding quotation marks at the beginning and end. Where a formula is repeated over a row or column, select only the first occurrence. Do not be concerned that the formulas do not fit into the cell space or that they run off the screen.

2) Copy each of the cells now containing the formula in string form to a convenient area below the working tables where they will be displayed in

full form. If wanted, add a suffix within each string to identify the cell from which taken or to match the formula identity produced by the Abacus formula printout.

3) Print the area containing the string formulas.

4) Either reload the original spreadsheet or return to each modified cell and remove the quotation marks from the formula strings before saving.

This method results in the formulas being printed with actual cell designations (ie A2+C3) in place of the relative cell designation shown by the Abacus PRINT command (ie [-2][-1]+[1][-1]). I personally think the former presentation with actual cell designation is easier to explain to anyone unfamiliar with spreadsheet terminology. This method was suggested in a comment by an earlier writer to *QL World*, whose name I have regretfully forgotten.

Keep up the good work. *QL World* is one of the very few magazines I do read from cover to cover - adverts included!

M Herbert L Keener
Wheathampstead
St Albans

Editor's notebook

Firstly, what do you think of our new page layouts? The nose-down headlines mean a more modern look; the varied positions of the intro lines and information boxes are easy on DTP, and (people tell me) many readers prefer ragged-right ('left-justified' in the new computerspeak) text, so that is what we are now using. Let me know if you like (or hate) it.

We were pleased to hear that Miracle Systems are run off their feet with work. "Please remind people that the QL is the only computer we work for," say Miracle, who have supported the QL almost from the start. Miracle demonstrate their wares at most Quanta meetings and selected All Formats Fairs.

Fred Toussi of Software87 has been in touch to say that selected software from Care, Jochen Merz and Cowo is to be available from Software87 any minute now. They will also be supporting the new QL emulator card for the Atari Mega STE and TT, giving high screen resolutions for the QL. More next month.

Mike and Sue Lloyd have, apparently, completed their move. The only snag is that one of the services (no names) insists on calling their new home 'plot 36', rather like a cemetery. Let us hope this grave dilemma is swiftly laid to rest.

Printmaster

Control your printer easily and successfully with Printmaster, says Henry Orlowski.

INFORMATION:
Program:
Printmaster
Price: £20
Supplier: Dilwyn Jones
Computing, 41
Bro Emrys, Tal-y-
Bont, Bangor,
Gwynedd LL57
3YT. Tel. 0248
354023.
Minerva and
Gold Card
compatible.

We all know that computer users have problems getting printers to do what they want. The regular letters on this subject published in QL World and Quanta testify to the widespread topicality of these problems.

Even when you've mastered the intricacies of your particular model you still need to remember the individual control codes, or look them up, and then send them to the printer through the driver or by opening a channel directly from SuperBasic - all that messing about with chr\$(27), "E" etc., is tiresome, especially if you move between different applications regularly and forever changing your printer set-ups.

One way round this is to put the codes directly into your document for the printer controller to pick up. That way any one set-up is only valid for that particular operation (unless you leave the settings in situ, of course). A better way is to use printer-driver-writing software, such as Hot Install, but this type of program only allows control over a select number of functions.

English

Or you can use *Printmaster*, a new text file printer control utility from

Dilwyn Jones. This is totally different from the alternatives mentioned because you don't need to know the control codes. The functions are simply listed in plain English for you to select for immediate transmission to printer. It's that easy.

Printmaster handles all your printer's functions once you have loaded the function driver file relating to your own printer model. A number of driver files are included in the package, already set up. These include the various Star printers, colour and 24-pin too; FX 80; Brother HR-5; and Serial 8056. If yours isn't included it's a relatively easy matter to configure your own driver, based on one that's closest, guided by the *Printmaster* manual and that of your own printer. It's also a simple matter to make your own function driver file load up as the default at program start-up time. You also have the ability to load any other function driver file subsequently once you are within the program.

The number of functions available depends, of course, on the printer. If you take as an example the Star LC10 mono, this lists 80 functions, including such settings as Margins, Print Modes, Character Sizes, Character Sets, Colour Selection, Line Spacing and many others. At the other end of the scale, the trusty 8056 only has 21 or so, but an 8056 owner I demonstrated the program to at a local meeting discovered a function he was still unaware of after several years, so this is a good reference guide by all accounts.

```

The cat
and the dog

The cat and
the dog

The
cat, and the dog

The cat and the
dog
```

```

the cat and
the dog chased each
other round the yard un
til
they saw a mou
se and a tiger whereup
n they ended up being
chased themselv
es in opposite direction
```

Information

In fact, in many respects the program would make an excellent basic printer manual, with all information conveniently available on screen and at the touch of a button. If you needed to know the particular code for any function or set of functions, you just select that function and ask for the codes to be revealed, note them down and then apply them accordingly.

The program can be run in either single-tasking or multi-tasking mode. As a single task it will run on a bare 128K machine. You can set up your printer as you want it for, say, wordprocessing and then quit and load up your wordprocessor.

Far more efficient and effective is to use it in multi-tasking mode and switch between it and your wordprocessor using Ctrl-C or under the control of a multi-tasking environment such as *Taskmaster* or *Qram*.

Upon loading the

Printermaster software you are asked if you wish to change the main default settings of the ser1 port and 9600 baud rate. Almost certainly you will not wish to. You are then presented with the main menu which lists all the available functions for the loaded function driver file. You are

other of the printer function files and/or save one with its own name or as the default, including one that you have just edited. Should you like the idea of a printed reference guide for your printer, you can also now get a hardcopy printout of all the available functions along with their

one selection and one keypress, with the entire sequence sent to the printer in one hit.

Function set

Let's have an example. Suppose you write a lot of letters, and you're happy for

that prior to printing out.

I would suggest that you either keep tabs on what the most recent functions sent to the printer are, and perhaps use Reset prior to sending new functions so that there is no confusion. The program does not recall or highlight the current valid functions sent to

"PRINTERMASTER"

↑ ↑ NO MORE FUNCTIONS ↑ ↑		
RESET PRINTER	CARRIAGE RETURN	BACKSPACE
LINEFEED	REVERSE LINEFEED	FORM FEED
REVERSE FORM FEED	EXECUTE n LINEFEEDS	TAB ACROSS n COLUMNS
UNIDIRECTIONAL PRINT	BI-DIRECTIONAL PRINT	NORMAL ZERO
SLASHED ZERO	CONDENSED PRINT ON	CONDENSED PRINT OFF
SENT-CONDENSED PRINT PITCH	EXPANDED PRINT ON	EXPANDED PRINT OFF
ONE LINE EXPANDED ON	ONE LINE EXPANDED OFF	UNDERLINE ON
UNDERLINE OFF	SUBSCRIPT PRINT	SUPERSCRIPIT PRINT
SUB/SUPERSCRIPIT OFF	EMPHASIZED PRINT ON	EMPHASIZED PRINT OFF
DOUBLE STRIKE ON	DOUBLE STRIKE OFF	ITALICS ON
ITALICS OFF	PROPORTIONAL SPACING ON	PROPORTIONAL SPACING OFF
DRAFT QUALITY PRINTING	NLQ PRINT ON	PICA PRINT PITCH
ELITE PRINT PITCH	HIGH SPEED ELITE PITCH	HIGH DENSITY ELITE PITCH
SET LEFT MARGIN (CHARS.)	SET RIGHT MARGIN (CHARS.)	SET TOP MARGIN (LINES)
SET BOTTOM MARGIN (LINES)	CANCEL LEFT MARGIN	CANCEL RIGHT MARGIN
CANCEL TOP MARGIN	CANCEL BOTTOM MARGIN	CANCEL TOP & BOTTOM MARGIN
↓ ↓ MORE FUNCTIONS ↓ ↓		
PRINTERMASTER MAIN MENU		
F1=See/Edit this FUNCTION	F3=Command Menu	ENTER=Implement FUNCTION
F2=See defaults	↑↑↑=Select FUNCTION (set with SHIFT or ALT)	ESC=Quit

able to cursor-navigate up, down and across. Each function is highlighted as you move across it, a ready indicator of where you are. If there are more functions than can be displayed in one hit, the window scrolls to reveal the rest. Helpfully, the navigation can be fast or slow depending on how impatient you are.

Revelation

You are now in a position to send any function to your printer with a single keypress. At this stage you can also ask for any function to reveal its codes and even edit them if you are setting up for a printer driver not supplied as standard. If you make a mistake, you can load up any

corresponding control code sequences. Useful stuff.

One of the most useful items is the ability to put additional functions into the function file, up to a maximum of 120. It's not that your printer can be made to do something it's not designed for, but you might wish to make a new file for a new printer with more functions than the basic one selected.

However, the main use is to enter additional function-sets. A typical function has around three control codes, but here you have the ability to enter up to 21. What's the use of this? Well, imagine that you can put more than one function within each function selection. This means that you can have a whole set-up sequence of several functions achieved by

those letters to be processed in the same way. The treatment may consist of:

- setting page length
- setting pica
- setting italics
- setting NLQ
- setting margins

Well, you could include all the above sequence within one function selection and call it, say, 'letter setup'. Whenever you wished to print a letter you would just hit 'letter setup' and the whole lot would be transmitted in one hit.

Then, when you transferred to draft listings you could have an appropriate set up called 'draft list' or similar and send

the printer but this is no great hardship.

Excellent

All in all Printmaster is an excellent, easy-to-use and very useful utility, a godsend to many struggling to get to grips with their printers. The question of value for money is a subjective one but I think most purchasers will feel the level of control provided over a difficult area of understanding makes it money well spent.

The manual is short and to the point, much of the program being self-explanatory. It also contains some very useful guidelines, hints and tips to make your life as comfortable as possible. Recommended.

3D TERRAIN

For the second time in recent months, Bryan Davies comes across a program which models data in three dimensions.

Despite having the same name, this program is not a version of the one reviewed in the May/June *QL World*. The earlier program came from the USA, whereas this one was written in the UK. Both are graphics programs which produce screen displays representing contours. While "terrain" implies it is the earth which is being displayed, it can be applied to many things. It is possible to "map" the state of the world's financial markets as a 3-D display, but that kind of activity is beyond the scope of the programs we have on the QL. The program introduction (in the instructions) mentions possible applications - air pressures, temperatures, rainfall, as well as the obvious one of actual mapping of land height.

The program is stated to be based upon a series of articles by Mike and Sue Lloyd in *QL World*. It was compiled with Digital Precision's *Turbo*. The Psion programs *Abacus* and *Easel* are required, the former to prepare data for input into *3D Terrain*, the latter only for its GPRINT_PRT printer driver file. You are clearly warned that this file has to be on the working disk (or cartridge), regardless of whether or not you intend to use it for printing, as the program won't run without it. A memory expansion of 64 KB or more is needed to run the program. About 6 pages of printed instructions are supplied, and there is an UPDATES_DOC *Quill* file on the disk to add information on recent changes.

Working

A working copy can be made using either the supplied clone routine or the WCOPY toolkit command. The *Easel* driver file has to be copied separately onto the working disk. If you have DP's Lightning display-accelerator, the LNG_GRAF_EXT and LNG_MATH_EXT files from that can also be copied to the working disk; Lightning can significantly improve the speed of the screen display changes. Some minor

changes may have to be made to the boot file, to accommodate operation from microdrive, and the use of Lightning. The instructions are clear on this, and the initial lines of the boot file are REMark lines which give the same instructions.

In connection with this, my initial attempt at running the program was unsuccessful. The boot routine assumes sizes for the two Lightning files, but these files, and their sizes, have changed over the years; you may need to change the space allocations in the boot file for the loading of these files. Once this problem had been sorted-out, the program loaded quickly and displayed the opening menu screen. This screen is shown in Figure one, with one of the sample files plotted on it. This is an actual print of the screen by the program, using GPRINT_PRT, not a screen dump from the Gold Card SDP command (normally used for review pictures). A laser printer was used, and it took about 2 minutes to print; the image would be essentially the same on a dot-matrix printer.

Parameters

Two parameters, Elevation and Angle, are available from the keyboard. Both have 20-degree ranges, elevation effectively raising the viewing position and angle moving it around the plot. This is sufficient adjustment to make the images obviously different, without being enough to allow the serious user to view data from wide-ranging viewpoints. Figures two and three show another of the supplied plots, at the two extremes of elevation and angle.

The image background can take one of three forms: the 3-D box with two opaque sides as shown in figure one, the same box with all sides transparent, or just the base of the box as used in figures two and three. A vertical scale is displayed with each background.

What comes out on paper is much the same as what is seen on the screen, in terms of image quality. The vertical scaling of the printout gives a slightly "long-faced" look to

INFORMATION

Program: 3D

Terrain

Price: £12.50

Supplier: CGH

Services, Cwm

Gwen Hall,

Pencader, Dyfed

SA39 9HA. Tel.

0559 384574

the images, and this increases the jaggedness of lines somewhat. Two options are provided for printout: one includes the menu down the right side, the other does not and is correspondingly larger.

As is normal with basic Epson FX-80 drivers, there is a lack of contrast in the printed image. Users with sufficient technical knowledge, and modern printers, might be able to develop drivers to give much better results (not just with this program). The *Easel* drivers must have been written in about 1983, and are getting rather long in the tooth; they don't make use of the higher-resolution graphics modes found on most printers now.

Sharing

The instructions indicate there should be an F9 (for PC-type keyboards) or Shift-F4 option on the Main Menu screen, to redraw the screen, but it wasn't there. This keying didn't cause that screen to be redrawn either, but it did perform this function when *3D Terrain* was run with other programs already loaded; switching in this situation caused parts of the 3D screen to be missing, but refresh was immediate with Shift-F4. There were no apparent problems when the program was sharing the QL with others. Users who habitually operate the QL in this way should check to see that toolkit and Lightning files are not being loaded twice, however.

Selecting F1 brings up the File Menu screen, from which you can select one of five sample plots, two of which are shown in the illustrations. Two of the samples appeared to be

the same. The device used for holding data files can be any one of the now-standard ones, such as hard disk; you need to have Toolkit II loaded to be able to select files from sub-directories.

What is likely to influence the potential buyer most is the facility for creating plots from one's own data. For this, Abacus is needed. It may be difficult to visualise what to put into Abacus to produce an image such as that in figure one. The data for this image is shown in the Figure four, as it appears on the Abacus screen. The instructions indicate that the data was taken from an Ordnance Survey map of North Yorkshire. The method for

obtaining the figures is to lay a piece of tracing paper marked with a 5 mm grid on top of the map, and note the contour heights at the grid intersection points. The

whole process took about four hours, up to generation of the _EXP file which is used as the input to 3D Terrain.

Twenty-five columns and rows were used, but the program can handle 50 x 50, giving four times the amount of data. On systems with either Trump Card or Gold Card, memory shortage will not be a problem, and screen resolution is the only limiting factor. Obviously, the user can scale the input data as required, to give either a fine-grained view of a limited area, or a coarser view of a larger area. The supplied file could be used as a starting point, with modifications being made in Abacus and tried in 3D Terrain, to see how changes in rows and columns affect the image.

Comparison

Judging by comments which accompanied the review disk, the programmers of this version feel it is appreciably better than the earlier program, but the two programs are significantly

different. The one reviewed in the May/June QL World has a range of built-in functions for changing supplied plots over a wide range, but it doesn't allow you to input your own data as figures, whereas the current one is a tool for representing any typical figure data from a spreadsheet in graphical form, but a relatively standard one. The samples are not as complex or varied as those with the US program of the same name, and the program doesn't provide the range of facilities for varying the plots.

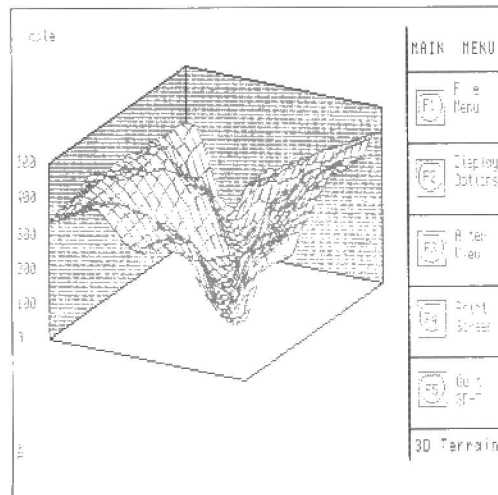
What this 3D Terrain does do is present the displays on the screen much more quickly, and allow the user to input his/her own data. Display of

plots of the complexity of that in figure one takes only a few seconds on a Lightning- and Gold Card-equipped QL. Using the same system, plots made with the program reviewed in the

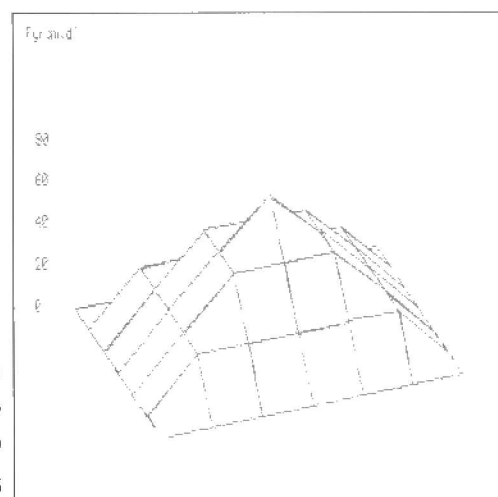
June issue took minutes or hours, but the level of complexity then was much greater, and it is not possible to make fair comparisons of program speed. To make real use of the program, it is necessary to have a reasonable feel for how to enter data into Abacus to produce images. How to present the data in two-dimensional form is up to you - the program will quickly give you a visual presentation of your data. The emphasis in micro software is currently very heavily on graphic presentation, and simulated three-dimensional modelling of this sort is a common business tool in business computing.

**abacus
screen
showing
data for
first
illustration**

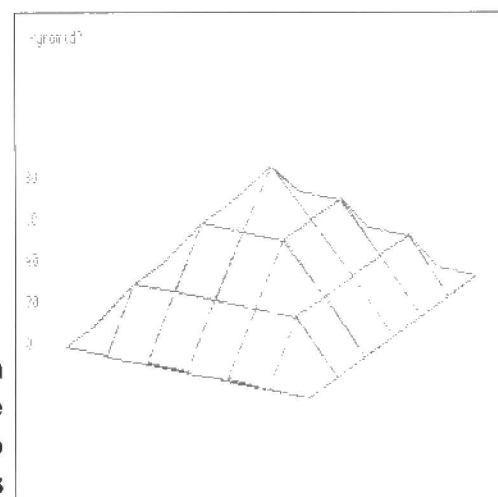
**main
menu and
sample
plot**



**elevation
and angle
set to 20
degrees**



**elevation
and angle
set to zero
degrees**



HELP	CURSOR	DATA & FORMULA	TEXT type "	COMMANDS
F1	press ←↑↓→			F3
PROMPTS	GOTO CELL	enter directly	followed by	ESCAPE
F2	press F5	& press ENTER	text & ENT.	ESC
A	B	C	D	E
1	1	1	1	1
335	338	345	350	355
345	349	355	370	370
360	360	370	382	387
370	370	370	375	375
383	385	380	371	360
395	397	386	372	356
400	400	393	378	358
382	392	393	381	362
355	370	372	370	354
333	341	346	350	350
315	310	315	314	313
308	289	297	270	275
325	322	319	295	263
340	335	325	305	263
353	344	333	310	268

MINERVA MkII

Qview's popular Minerva rom has now moved from numerous Mk I versions into Mk II. Minerva user Ian Bruntlett tries it out.

Minerva MkII is the latest version of Qview's replacement operating system for the QL. Minerva is supplied on an eeprom chip and a small circuit board, along with a manual and a utility disk. The difference between a Minerva MkI and a Minerva MkII is the difference in the printed circuit boards. The MkI is a simple carrier for the eeprom, while the MkII has an I2C bus, and a battery-backed clock and ram. For the review I had to put the old roms (JM, JS) back in my QL to test against Minerva. As far as I am concerned, nostalgia is dead - I won't willingly return to either JM or JS.

Fitting

Minerva MkII was easily fitted. I used a Philips screwdriver and the chip removal tool from Qimi. If you are fitting the new rom to a Gold Card QL, you don't need to remove the battery from the Gold Card, according to Miracle. A Gold Card rom of 2.26 or later is needed to work with Minerva MkII. It has worked without problems in a QL with Qimi plus RTC and the Gold Card. It has also been used on an unexpanded QL with no problems, and with a Trump Card QL.

Normal users

Non-technical users of the QL will find their machines faster with Minerva installed - but only if the programs used on the QL depend on the operating system a lot. A program such as *Abacus*, which has its own routines for the bulk of its work (such as maths, and maintaining the spreadsheet) will not benefit much from Minerva. My most demanding application on my QL is compiling C programs -

this makes demands on the processor, storage devices, and so on. Compiling the same, small library with a Gold Card and C68 3.02 (beta) took 69 seconds on a Gold Card, 58 seconds on JS and 55 seconds on Minerva. This shows that a Minerva QL, given the same hardware and user workload, will usually finish before QLs equipped with JM or JS roms.

Anything that uses the QL's graphics routines for drawing pictures is significantly faster and more accurate on Minerva than on JM/JS. When I tested the routines for printing text on screen, I was astonished to find JS is about 10% faster than Minerva. Multiple runs of the test had been taken, each taking over an hour on a QL with a Gold Card, so this was not a single fluke result. The same test, with *Lightning Special Edition* installed, had Minerva slightly faster than JS, presumably because the TRAP handling has been speeded up so the Lightning SE routines are called quicker on Minerva.

Faster SuperBasic

SuperBasic programs run faster on Minerva according to a benchmark program from the Quanta library. The improvement is not to be sniffed at - but if speed is required of a SuperBasic program then the programmer should consider a compiler or assembler extensions.

If you want to run two or more SuperBasic programs at the same time, or run SuperBasic programs while still having access to the SuperBasic command line, then you need Qview's MultiBasic. MultiBasic allows extra SuperBasic jobs to be run on a QL, and this useful facility is not available on any other rom.

The QL's reset prompt appears more quickly as the memory test has been made more thorough and faster. Extra information is given - rom version, current time, memory capacity. There are some new options - a memory shrink to 128KB, ignoring plug-in roms, enabling the second screen - all available from the F1/F2 prompt. Some

early software was written so badly that it won't run on an expanded machine or with certain roms, so these options can be really handy. Minerva will wait for a keypress at the F1/F2 prompt for up to 15 seconds - then it assumes you want F2, and loads your boot program.

With the MkII Minerva, though, a command can be stored in the battery-backed ram to be entered at boot-up. If you have F1/F2, etc., within a command string it can be hard to use the other options, such as the 128K reset. In those circumstances the best thing is to use the configuration program provided to remove the boot string.

The keyboard routines have been enhanced amongst a list of changes; three stand out. The soft reset, Ctrl-Alt-Shift-Tab has been dealt with. Ctrl-C is normally the key to switch jobs on the QL, but Conqueror changes it to Ctrl-Shift-C. Some other programs change it to impossible keypresses. To get round this, Ctrl-Alt-Enter is a Ctrl-C that can't be messed around. If you only rarely use the accented characters on

INFORMATION

Product: Minerva MkII (RTC), version 1.93
Supplier: Tony Firshmann Services, 12 Bouverie Place, London, W2 1RB. Tel. 071 724 9053
Price: £65 (£60 with proof of Quanta membership). To upgrade from Minerva Mk I to Minerva Mk II, send to TF Services: All of Minerva Mk I: the manual, Minerva eeprom and board, plus a blank disk and £35.

the QL, you'll find it hard to remember the various shifts required. Now you just use Ctrl-Enter-e-/ to get an accented e.

SuperBasic programmers

Writing programs in SuperBasic is much more comfortable with Minerva keyboard input for EDIT, AUTO and INPUT have extra keypresses for delete/move to start or end of a line. When a syntax error is detected on a command line, the cursor is moved to the last letter SuperBasic understood. This only applies to EDIT and INPUT - this feature isn't used by *Toolkit 2's* ED. Although INPUT has been improved, it still isn't possible to get SuperBasic to edit the contents of a variable that is being input - for example, INPUT (!q\$) would not be accepted by roms that don't have the facility, but on others would allow the user to edit the contents of q\$.

If you are using SuperBasic programs to handle data in conjunction with the Psion programs (I do this with surveying data), then MultiBasics are a handy way to have two SuperBasic programs loaded in memory ready to go. In a working environment where the demands on the programs can easily change, it is not always convenient to have all your procedures in one SuperBasic file. Using MultiBasics on Minerva, I can have instant access to an "Angles to Bearings" module and an "incorporate elevation and point info", module without having to have them in the same program.

Implemented on the JS rom and polished on Minerva, WHEN ERROR is an error-trapping routine run by SuperBasic when an error occurs. If you want to trigger an error deliberately the best method is to use ERT from hot_ext. Placing "ERT -7" would simulate a "not found" error. Some caution is needed when writing WHEN ERROR clauses as its possible to get SuperBasic to lock up. Once you have a WHEN ERROR clause working, it is stable.

Help with debugging

To help with debugging, WHEN clauses can be set up to be run when a variable is modified. This is of limited use as it is not triggered by an INPUT, READ or BP.LET. A WHEN STEP or WHEN LINE, triggering a routine either at the end of every statement or every line, would be able to track the contents of a variable more thoroughly but much more slowly. Having used the Easy Ptr toolkit, which silently modifies a parameter variable to signal an error, I know from frustrating experience that a WHEN LINE would be a lot quicker than I am at spotting the changing of a variable.

More help is provided with a small trace toolkit that will trace or single step a range of lines. The information provided is line and statement number plus a little symbol. The manual doesn't say what the symbols mean, look at the table on TRACE in the ASM chapter - the missing column of symbols are .:=;fR starting at TRC.STST.

In Minerva there are additions to these keywords, some of which are overwritten by existing software: ABS, ATAN, BLOCK, DATE, SDATE, MODE, PAUSE, FILL (bug fix lost when PTR_GEN is loaded), OPEN/_NEW/_IN SBYTES SEXEC (overwritten by Toolkit 2), MODE, PEEK, POKE, RENUM, RESPR, SCALE, VERS, WINDOW.

The various PEEK and POKE commands have been greatly improved, but in a slightly odd way. The facility to PEEK and POKE the system variables or basic variables safely is very handy and POKE, extended to take a list of values, is more convenient. But there is still no ability to handle strings easily and although writing a PEEK\$/POKE\$ in SuperBasic isn't hard, it is inconvenient and slow. A PEEK\$(address [,maximum length]) function would be very handy. One change, the ability to handle words and long words with odd addresses is, usually unhelpful because whenever I try to use words/long words at an odd address I am usually doing something wrong. So an

error message at that point would be helpful.

CALL not extended

Curiously, the companion to PEEK and POKE, CALL, has not been extended. This is probably because CALL gets rewritten by Toolkit 2, as in earlier QL roms it was flawed. Some keyword which doesn't get patched by existing Toolkits needs to be available to call Qdos traps. This would mean that the facilities provided by the extended OPENS, SBYTE and SEXEC could be provided by calling the Qdos routines directly. Given that CALL and RESPR are taken over by various toolkits either a new command name will need to be used or the PEEK_L command could clumsily take the load. Perhaps this would be the result:

One way of doing an extended call:

```
d0 = PEEK_L(addr,  
register_array  
[,register_array] )
```

One way of doing a Qdos trap:

```
d0 = PEEK_L(!trap no,  
register_array  
[,register_array] )
```

And a Qdos trap with addresses relative to A6:

```
d0 = PEEK_L(\trap no,  
register_array  
[,register_array] )
```

Note - these are just suggestions, not working examples!

The SuperBasic language has been extended so that FOR loops and SElect constructs can work with integers and strings as well as the normal floating point variables. Also, when constants like 1 are stored in a program, they will be stored in integer form taking one or two bytes instead of the six bytes of floating point. The result is that integer calculations involving constants are quicker. This change can upset SuperBasic compilers,

so it can be switched on and off. QLOAD and QSAVE handle this badly, and should really check to see if the tokens in the file are compatible with the "token mode" of SuperBasic when loading, and give an error message if the modes are incompatible.

Number slicing

Slicing used to work only on strings. Now it will work on numbers and the results of functions, but beware of trying to slice a number from an array. SuperBasic can lock up.

Machine code programmers will have fun with Minerva. There are so many places where routines have been improved or added. Writing toolkits for SuperBasic should be easier with vectors for getting channel numbers, getting a new channel number, getting string parameters without quotes, adding new keyboards, moving/clearing memory, starting MultiBasics and new maths vectors. The disadvantage is that the information on the routines varies from sparse to complete, and there is little info available on which version of Minerva a vector was first usable on. This is important because a program that tries to use the Minerva vector BP.CHNID should stop with an error on a JS QL or an early Minerva rom, instead of just crashing the machine. Qview never mention any method that they will guarantee to detect Minerva. Looking for 'JSL1' in the new system variable sx_basic would seem to be an answer, but sx_basic was put there so it could be changed.

The one change I don't like is that to BP.INIT, which rejects bad names (I can live with that), but the user gets no warning that the toolkit they have just loaded hasn't been completely linked in. An error message giving the offending name to the user would be much more helpful than silently ignoring the rest of the toolkit.

A new set of system variables allow various routines to be replaced by user routines.

Incompatible software

If you rely on zombieware - programs written a long time ago that are no longer updated - then you should try them out on a Minerva QL before you buy Minerva. A local Quanta subgroup would be a good place to find a Minerva QL. On the whole, if you can run your programs with PTR_GEN in the machine, they are likely to run on a Minerva QL. Several bodge routines are provided by Qview to get programs running on Minerva.

Some popular programs need some adjusting-for. Old Qliberated software will need to be patched, as will version 2.06 of hot_text, pre 1.7 versions of QLOAD/QREF and old versions of Turbo Toolkit. The system enhancements have to be turned off to allow Solution and Conqueror to format a disk and there is a slight hiccup when using SDUMP on Minerva.

Minerva MkII hardware

This gives a real-time clock with a battery backup that will not be affected by QL crashes. The clocks on the Gold Card and the Qimi plus RTC interface can both be corrupted by a severe crash. The Minerva MkII clock can only be corrupted by a bad serial command on the I2C bus. If you aren't experimenting on the I2C bus, then Minerva MkII is a perfect way to have a secure clock within your QL.

Besides the real-time clock, there are 240 bytes of ram used for configuration info, a boot string and a user area. A config program is provided and works despite well despite its primitive display.

The configuration info includes the ability to ignore individually plug-in roms, network station number, whether or not system/basic enhancements should be turned off and the types of devices attached to SER1,

SER2 and PAR.

The I2C bus is a serial bus that is capable of driving multiple devices with multiple inputs and outputs. The advantage is that there are fewer connections between

chips, making it more reliable in production (fewer connections to go wrong) and cheaper (smaller board to produce). The disadvantages

are that it is slow for a computer (but fast for serial communications) - about 12.5k/second and it requires special chips designed to work on the I2C bus. These aren't as easy to get hold of as some other chips.

Conclusion

From a technical point of view, the Minerva eprom should be in every QL. Viewed

as a whole, there are a few deficiencies. The manual is based on the text files sent out with Minerva. There are few diagrams, and the chapters aren't numbered, hampering its use as a reference guide.

Some changes have been made to keep the manual up to date, but nothing has been done to take advantage of the book format.

When the cost vs. benefits are considered, it is necessary to ask: "Is it worth it?". At £65, Minerva MkII is only a worthwhile move for current Minerva

users if they want either a high quality battery backup for the system clock, or are going to do some serious hardware development with the I2C bus. All the information I have points, to my mind, to Minerva MkII being a little over-priced, especially if considered as an ungrade. Minerva MkI, at £35, has all the facilities of MkII except the I2C bus, the battery-backed clock and the configuration ram.

T
TF Services
F

MINERVA

The ULTIMATE operating system upgrade

MARK II MINERVA now with battery for 256 bytes: ram, CRASHPROOF clock & interface (I2C) for robots etc. Can autoboot from battery ram.

Other features (common to MKI/MKII):
Autoboot on reset/ Multiple Basic/ Faster scheduler/graphics (180° of Lightning)-string handling etc/ WHEN ERROR/ Second screen/TRACE Foreign sybd drivers/ 'Warn' fast reset etc

MKI to MKII UPGRADES: up to 1.79 - £5. Rest FREE (send row #PROTECT LEGS) and disk or 2 mds)

Original MKI.....£40 MKII.....£65

ALL VERSIONS FULLY GOLD CARD COMPATIBLE

HERMES

A replacement QL co-processor (IC 8049 IPC)

Do you get keyboard bounce?
Do you find fast serial input unreliable?
Do you want to run an MSI 14400 bps modem?

If you can say YES to any one of these, then this is a product you have been waiting for

- Gives up to 19200 bps RELIABLE serial input
- Independent baud rates
- Stops keyboard "bounce" (unwanted repeat chrs)
- Improves "fuzzy" and "random" sound
- Provides extra input and output lines
- Key click
- Reset/INT? invoked more safely

FITTING is a simple job. Simply remove the QL top (8 screws) and replace the chip marked '8049' next to microdrive 1.

PRICE (including manual & software) £25

BULLETIN BOARD

OBBS - The first UK QL scrolling bulletin board. Now even larger with international ECHOmail and 10mb of files (including the WHOLE King James' bible!)

071-706 2379 W21/23/22/22bis 24hrs

Qualsoft

Viewdata/VT52/VT100 QL TERMINAL EMULATOR

Multitasking for electronic mail, PRESTEL etc, UET download. Compatible with Tandata Uconnect QL modem/adaptor. Use Uconnect RS232C to drive unbuffered modems. Comprehensive RS manual.

Software (3.5" or ndv) & manual £30

FILE TRANSFER to transfer text and programs error free (XMODEM) between computers

Available for IBM compats, ATARI ST & Sinclair QL Connects to Psion range with RS232C comms link. Software £7.50 per m/c. Cable £10 (name 2 m/cs)

All prices include VAT and postage in the UK

QL SPARES

Faulty QL boards (excl plug in ICs).....£9

Keyboard membrane.....£8 Circuit diags....£2

68008 cpu.....£10 1577 PAL.....£3

1M rom set.....£10 Power supply.....£12

8382 ULA.....£10 8381 ULA.....£10

8849.....£8 MDV ULA.....£12

Other components (sockets etc).... please PHONE

QL REPAIRS

QLs tested with Thorn EMI test rig and ROM software

(MDV hardware extra if required)

£27 - 6 month guarantee

MAIL ORDER ONLY

12 Bouverie Place, London W2 1RB (tel: 071-724 9053)

SCROLLING BBS FOR ORDERING/QL HELP AND FAX - 071-706 2379

★
SINCLAIR QL
★

STOP PRESS

Serial 3 Button Mouse

Remote Infra Red Mouse

Externally fitted into SER2, comes complete with interface and software. No Internal fitting required, 3 button mouse gives 5 more functions than the QIML

£45

£55



A SERIAL MOUSE FOR THE QL AND SER Mouse the Mouse Driver

The SERIAL MOUSE SYSTEM enables a mouse to be used with the QL by simply plugging the mouse into the SERIAL 2 Port, and running a BOOT program. Common to all Mouse Systems, programs with a Pointer Environment are required.

Four Packages are on offer

1. 3-BUTTON MOUSE, SER Mouse 3.5 SOFTWARE, D-9pin to QL ADAPTER FOR THE SERIAL PORT. INSTRUCTIONS. **£45**

*1A. AS FOR 1 ABOVE, but including PC MOUSE DRIVER DISK. **£12 extra**

2. CORDLESS (INFRA-RED) 3-BUTTON MOUSE, SER Mouse SOFTWARE, D-25pin to QL ADAPTER FOR THE SERIAL PORT. BATTERIES & INSTRUCTIONS **£55**

*2A. AS FOR 2 ABOVE, but including PC MOUSE DRIVER DISK, and 'VISUALIZER' a PC Mouse Painter Program on 3.5 and 5.25 disks. **£18 extra**

* While stocks last.

NB. We still have lots of QL's, 1 & 2 Mb disk drives, peripherals and components. See previous adverts or contact us for details

EEC LTD

18-21 MISBOURNE HOUSE,
CHILTERN HILL,
CHALFONT ST PETER,
BUCKS
SL9 9UE

Tel: 0753 888866

Payment terms:
CWO, Access, VISA et cetera
Cheques - allow 10 days
Delivery:
Postage - £3
Product subject to availability. E.A.O.F.

Fax: 0753 887149

SER QDOS EMULATOR

**Simon
Goodwin runs
QL programs
on the
Commodore
Amiga.**

The Amiga Qdos emulator is a remarkable program: unlike most emulators, it is faster than the machine it impersonates, yet it needs no extra hardware. The program is in the Public Domain, and runs on all models from the Amiga 500, which has sold over a million units in the UK, to the latest 68030-powered Amiga 3000.

The Amiga is an American computer, launched in 1985, a year after the QL. It uses Motorola's 16-bit 68000 processor, plus lots of extra hardware designed by former Atari staff, including Jay Miner, who designed the Atari VCS, Atari 400/800 micros, and many arcade machines, and set up a separate firm to develop the Amiga.

The Amiga was very sophisticated for its time, based on several fast, complex chips. Custom chips let it manipulate memory with a 'blitter', read and write disk tracks, and replay stereo sound samples, all without disturbing the main processor! The Amiga was designed as a games machine, but the 68000 and custom chips are flexible enough that it is a powerful all-round computer.

Commodore arrives

When Atari ran short of Warner Brothers' cash in 1984, the Amiga project was snapped up by arch-rivals Commodore, who hired UK software house Metacomco to provide it with a 'serious' operating system. Metacomco had demonstrated their ability to program the 68000 family with their 1984 QL products, including compilers for BCPL and Lisp, and a 68000 assembler package.

The result was Amigados, slow bulky compiled BCPL with an old-fashioned command-line interpreter that loads a new program for each command. This works if you have a hard disk, but it is painful when using floppies. Alternatively you can use 'Intuition', a simple mouse-based desktop system, but this still needs software from disk and only reveals '.icon' files.

Early Amiga systems came with Metacomco's ED, from the QL Assembler Development Kit, but since version 1.3 Commodore have included *MicroEmacs*, the Amiga version of the excellent Public Domain editor reviewed in August's *QL World*.

The Amiga's 68000 architecture supports up to 10 megabytes of ram, or even more if you upgrade the processor to a 68020 or 68030. Amiga games often make far better use of the custom chips than the Commodore Amigados software. Most games turn the operating system off and take total control of the machine. QL programs are more likely to multi-task than Amiga titles!

Pedantic software

The Amiga consists of brilliant hardware, with slow and pedantic operating software. The QL consists of simple and rather eccentric hardware, with a fast, concise operating system. If only the QL operating system Qdos could run on the Amiga hardware...

The Amiga Qdos emulator was developed by Rainer Kowallik of Berlin, and since improved by QL and Amiga enthusiasts. The first version was 3.03, released to the public domain in 1988; this ran *Quill* and many other QL programs but had serious bugs, including a tendency to crash when memory ran short, plus slow and incompatible disk handling.

Version 3.10 arrived soon after, with corrected disk routines and improvements to take advantage of expanded memory systems. The latest version, 3.20, is being tested as I write.

Amiga ports

The Amiga has mouse, serial and parallel ports built in, and these are supported by the emulator. The mouse driver in version 3.10 is jerky if you move the mouse too fast, but smooth if you move the mouse more slowly. BEEP, MDV and the Sinclair NET devices are not supported, but the real-time clock works, and the emulator lets you open

channels to SER, FLP, PIPES and PAR, just as you might on a QL.

I found that SER on version 3.10 could receive data from the QL reliably at up to 9600 baud, but lost occasional characters when receiving at 19200 baud. I linked the machines with the QL serial printer lead, and paid a pound at an All Formats Show for an RS-232 'gender bender' to adapt Sinclair's 25 pin D-type plug to fit the Amiga connector.

I am told that the first implementation of the SER driver for Version 3.20 transmits successfully to the QL at up to 1200 baud, and manages 19200 baud when talking to an Apple Macintosh.

The blitter

The QL screen emulation is the most ingenious part of the Qdos emulator. It uses the Amiga's 'blitter' - a custom co-processor which can transfer and manipulate memory shared with the main 68000 processor.

Such shared memory is called 'chip memory', and normally comes at the start of the Amiga memory map. The first A500 computers were limited to 512K of chip memory, but later models such as the A500 Plus and A600 allow one or two megabytes.

A standard one megabyte A600 gives the same memory availability as an 896K Trump-Card-expanded QL. If you double that with an internal board, you get two megabytes available to Qdos, as on a Gold Card.

The Amiga's 68000 has a 24-bit address bus, so up to 8 megabytes of further memory can be fitted higher in the Amiga memory map. This is called 'fast memory' because it is only accessible to the main processor; the other Amiga chips cannot slow down access by using it at the same time as the 68000.

Red and green bytes

The QL display area is conveniently held in chip memory, at 131072. However

the QL keeps red and green bytes alternately in Mode4 display memory, while the Amiga display circuits expect two 16K 'bit planes' for red and green. The blitter is used to transfer the QL display contents to the Amiga bit-planes, so that any change in the QL display memory is reflected on the Amiga screen.

The Amiga high-resolution display allows up to 640 pixels per line, but Amiga Qdos only uses 512, like QL Mode4, so the emulator only uses four-fifths of the available screen area. Many monitors - such as my Philips CM-8833 - have a 'horizontal width' control which you can tweak to compensate for this.

The emulator lets you set the amount of time available to the blitter. If it is running too slowly, scrolling displays of white ink or paper-flicker, as the blitter updates first the red then the green bit-plane. If the blitter is running too fast, the performance of the 68000 suffers needlessly.

Early versions of the emulator expect you to set the blitter priority with a POKE. For instance POKE 164082,127 virtually eliminates flicker, but reduces the bus bandwidth available to the 68000 to 1840 kHz, enough time to read 1,840,000 bytes of code or instructions.

The default value of 20 lets the 68000 run at full speed, around 2834 kHz. A reasonable compromise value is 40, which gives 90 per cent of top speed. The figure also shows comparative timings for the 16-bit Thor XVI and QL-based Thor I. Expanded QL timings depend on ram speed, and may vary between the 128K QL and Thor I values. The example matches a typical 512K Expanderam (old-fashioned though that now is).

Installing

I bought my A500 Amiga for £150 at a Radio Rally. It has 512K of chip memory and 512K of expansion, at \$C00000 near the top of the memory map. The screen, bit planes, track buffers and system variables live in the first 512K. Qdos needs a contiguous area for tasks,

SuperBasic and the common heap, so these load into the top 512K, leaving about 500K free for SuperBasic.

On such a configuration much of the system variables area consists of a massive slave block table, to stop Qdos trying to use the absent memory between the two 512K chunks, but some spare chip memory remains, and can be used to hold toolkits and device drivers.

The listing searches the memory allocated by version 3.20 of the emulator. FIND_CHIP locates spare memory, and INSTALL uses it to hold Qdos code files. This is not useful in Amiga systems with all chip memory, but it may suit common configurations with memory split into two chunks. The emulator can use any area of memory for Qdos, if the parameters to the L_QDOS command are set appropriately; compatibility demands that the rom image, screen and system variables occupy the first 256K.

Compatibility

The 68000 processor is almost completely compatible with the QL's 68008, as the Gold Card and ST emulators have proved, but one quirk of the Amiga design causes problems for Qdos. The TAS instruction (Test And Set) is intended to read and write a byte in one indivisible step. Motorola use an unique 'read-modify-write' cycle to do this, but the Amiga hardware disrupts that; the custom chips use the memory bus between the read and the write, so the result is not correct.

Rainer Kowalik has written a SuperBasic program that

disassembles QL code and replaces any TAS instructions it finds. This TAS replacer requires Toolkit 2 commands, and is quite slow. It takes seven and a half minutes to scan the 60K Turbo PARSER_TASK, and replace one TAS instruction, used to handle the exclamation mark separator in PRINT - but you only need to run it once.

Patched programs will not run thereafter on a QL, as the emulator replaces the TAS instruction with an opcode starting with the hex digit 'F'. These 'F' line codes crash the QL, because Sinclair did not include an exception handler for vector 11, but they are intercepted by Amiga Qdos and treated as TAS instructions.

Serious programs

The Qdos emulator runs the majority of serious QL programs, but there are a few incompatibilities and bugs which even the TAS replacer cannot solve. Most of my tests have used Qdos 3.10, but version 3.20 should be even more compatible, once pre-release bugs are fixed.

Some QL titles are copy-protected and need a 'key' on microdrive; these will not run unless the copy-protection is removed, and may still give problems. Many games need Mode8 and QL sound to work properly; some, like Psion's

Match Point, work properly but are too fast to play on the Amiga. Psion Chess is microdrive protected, and uses TAS instructions; worst of all, it writes to memory in the QL rom area, so it crashes the emulator! The CGH game Uncle Loony's Legacy works fine on Qdos 3.10, apart from the lack of sound, so other games developed with ACT should also be compatible.

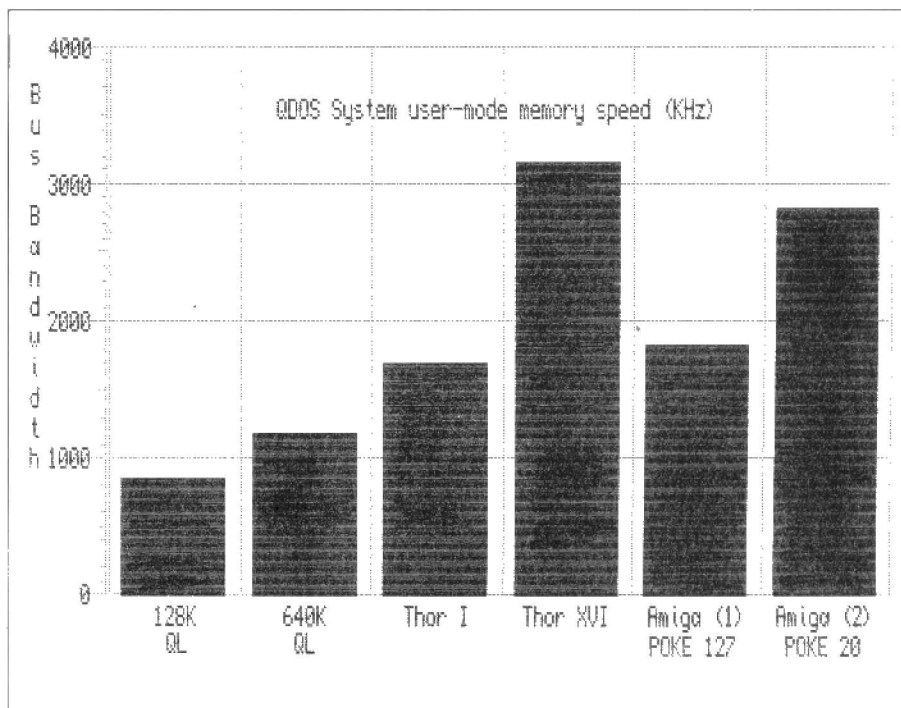
The QL suite of Quill, Archive, Abacus and Easel works perfectly; you may like to turn up the blitter priority when scrolling bold text in Quill. I have also successfully used Thor/OPD Xchange, as patched for the expanded QL, and hear that all but the earliest versions of Text 87 are correctly emulated.

Patching Toolkit 2

Toolkit 2 uses TAS instructions in the ED screen editor, and must be patched to run on the Amiga. Once patched the rom version can be loaded into memory in the usual place:

```
LET A=49152
LBYTES FLP1_TK2_CODE,A
CALL A+PEEK_W(A+6)
```

This uses the 16K area from 49152 to 65535, so the Toolkit does not take up any Qdos ram. This does not work in version 3.20, which stores emulator link routines after the



Qdos rom image at 49152. However you can load the Toolkit code into RESPR space, or perhaps load the code elsewhere in chip memory, using the program I have listed to find a space.

Turbo Toolkit and *DIY Toolkit* commands work perfectly on the emulator, with no need for patches. Arced has no TAS instructions and works well, without the keyboard over-run apparent when the arrow keys are held down in other Qdos editors, but for some unknown reason F1 and NEW fail to open extra windows, when run on version 3.10 of the emulator. I also successfully tested XREF, DiscOver and Basic Reporter from Dilwyn Jones Computing.

Ramdisk caution

The emulator documentation warns that you must not format ramdisks when using Tony Tebby's RAMPRT driver on the Amiga. The dynamic ramdisk code works, but the static version goes wrong. A slower Public Domain ramdisk driver is on the emulator support disks.

The QJump Pointer Interface is correctly emulated, as long as you patch the TAS instructions in PTR_GEN. Prospero's *Pro Pascal* and *Pro Fortran* compilers need a rom cartridge to run on a QL, but can be used on the emulator as long as a run-time file on the emulator support disk is loaded before each pass of the compiler.

Ironically, Metacomco's vintage ED, BCPL and Lisp compilers work fine under the emulator. Computer One Forth also works, but their Pascal compiler will not run on Amigas with expanded memory.

Q-Liberator works under the emulator, as long as you patch the TAS instructions in its compiler library. Turbo tasks are unreliable on early versions, but compiled tasks and version 2 and 3 of the compiler work on version 3.20.

TAS caution

The TAS replacer cannot always identify TAS instructions correctly, as the

same codes might be used for other purposes. You should not run the TAS replacer on Devpac, for this reason. The un-patched file works.

QL programs which expect Mode8 will run on the emulator, but the extra colours appear as Mode4 stipples. The new emulator adjusts the palette when Mode8 is selected, giving a better match between the stipples and expected QL colours.

The Amiga's colour palette hardware allows 16 levels each for the red, green and blue beams that generate a colour TV or monitor display. In combination, these settings give a range of 4096 colours. By default the emulator palette settings correspond to the black, white, red and green of a QL Mode4 display, but you can assign any colours you like, without disturbing programs.

Easy colourchange

It is trivial to change the colours from SuperBasic. Just write a new value, from 0 to 4095, to the corresponding Amiga hardware register. The word at address 14676352 controls the colour of the border and PAPER 0. Normally this holds zero, but you can replace black with any other colour using POKE_W. For example:

```
POKE_W "14676352", 1
```

sets the background colour to the darkest shade of blue. The inverted commas are needed around the address, because it exceeds SuperBasic's seven digit display limit.

If you enter a program line containing this value it will be listed as 1.467635E7, which is accurate to seven digits, but not accurate enough to correctly identify the address. The quotes ensure that SuperBasic holds all the digits, and coerces them from text to binary as required. Direct commands to POKE the palette work without the quotes, but program lines need them, or the address will be wrongly stored when the line is edited.

Other shades of blue correspond to values from 2 to

15, in order of ascending brightness. The next four bits determine the level of green, with 15 shades ranging in value from 16 to 240. To mix blue and green, add the two values; thus 255 gives the brightest shade of cyan. The next four bits determine the red component, from 256 (red level 1) to 3840 (red level 15, no blue or green component).

Other colours

The other colours are controlled by hardware registers in subsequent words. The colour of INK 2 is held at 14676354; try POKE_W "14676354", 10 to change red areas of the display to blue.

INK 4 is controlled by the third register, at 14676356. Use POKE_W to set this to 120 if you fancy a change from the usual shade of green. The last register used by the emulator is at 14676358, and controls INK 7, normally white. Try POKE_W "14676358", 4080 to fade the blue component.

All these values are easy to understand if you view them in binary, using the Toolkit BIN\$ function. For instance, PRINT BIN\$(4080, 12) gives the pattern "111111110000". Each group of four bits corresponds to a different colour beam. Note that these are hardware registers, not memory addresses. You cannot read the settings with PEEK_W, and attempts to do this are likely to change the contents!

Disk tricks

The Amiga disk controller can read an entire 5.5K track from disk to memory while the processor gets on with other work. Unfortunately it stores the MFM (Modified Frequency Modulation) code that is written on the disk, rather than the Ascii data sectors expected by Qdos. QL disk controllers translate MFM automatically, but the Amiga needs software to extract the nine QL sectors from the MFM track and pack them back together before they can be re-written.

This was the weak point of the original Amiga Qdos emulator. Version 3.03 took about 8 minutes to format a

QL disk, and still got the checksum wrong, so that QL disk controllers would not be able to read the resultant sectors. This meant that a disk was rejected by the QL once the Amiga had written to it: the directory was unreadable.

Version 3.10 fixed this problem, but still took about six minutes to format a disk. If you have two drives, it's much faster to use the new DISKCOPY command, which will copy a complete 720K disk from one drive to an unformatted disk in another drive in around two minutes.

Disk formatting

The standard QL format allocates disk sectors in groups of three, from alternate sides of the disk. Thus the first three allocated are sectors 1, 4 and 7 on side 0, followed by the same sector numbers on side 1, then 2, 5 and 8 on side 0, and so on.

This arrangement penalises the QL emulator, which must read a complete track at a time. Disk access is three times faster if the sectors are not allocated from alternate sides! Disks formatted with the emulator use a modified interleave table, so the emulator is much faster at reading its own format.

QL systems can read the Amiga interleave at almost full speed, so there is no problem with backward compatibility. The DIY Toolkit FAST FORMAT from Volume D or QL World November 1990 does not interleave sides so it works well on both QL and Amiga.

Version 3.20

I called Rainer Kowallik in Berlin to check the latest progress of the QL emulator. He has tested version 3.10 on a 68030 Amiga, and written a special TAS replacer which is compatible with chips like the 68882 and 68040, which use the 'F' line instructions for their own purposes.

Rainer's postgraduate physics research has led him from Amiga to Sun workstations, but further development of the emulator is now underway in the UK, in the very capable hands of

Mark J Swift, winner of the *QL World Artist of the Year* competition a few years back.

The first notable advance is much improved disk access in version 3.20. The development release runs the 720K DISKCHECK program from QL World November 1990 in one minute 42 seconds, vs. two and a quarter minutes on the QL. It keeps track of disk changes, so the directory appears at once if it is held in slave blocks, when the QL would have to turn the drive on briefly to check that it had not changed.

Some drives do not implement the disk change signal. The new command MOUNT tells the emulator to make a drive available to Qdos, while the command DSKCNG signals a disk change to the system even if the hardware will not.

Format time

The time to FORMAT a Qdos disk is now down to an acceptable two minutes and ten seconds. Users with more than one drive will probably still use DISKCOPY. The Amiga is fastest when it is copying raw tracks, as it does not need to translate information from MFM code into Ascii, or vice versa.

Version 3.20 keeps separate track buffers for each drive, making access to multiple drives very much faster. Other improvements by Mark Swift mean it now uses the hardware timers - rather than the 20ms frame interrupt - to move the disk drive heads at top speed.

New blitter code can update the entire 32K screen up to 25 times per second; version 3.20 interrupts the 68000 more than its predecessor, at its top rate for display updates, but you can set any rate you like with the command SCR_PRIORITY.

The top rate uses SCR_PRIORITY 8,1 but emulation is almost twice as fast if you select the default 4,1 which updates one quarter of the screen every 20ms. SCR_PRIORITY 2,1 lets the 68000 processor go flat out, but causes the same red/green flicker as the original emulator.

JS compatible

Version 3.20 aims for compatibility with Sinclair's JS rom. The new design implements Amiga devices in separate files which share memory rather like QL expansion roms. This should make it easier to run Minerva, Argos or other roms under the emulator. POKes to MC.STAT at 98403 and Ctrl-Tab already direct the blitter to use the second screen, although the current Amiga emulator keeps its system variables in that area, like Sinclair roms.

Version 3.20 assembles under Amigados, whereas earlier versions used GST's QL Assembler. The Amiga Public Domain assembler works faster and is included on the source disk for version 3.20. It is A68K, a relative of the PD assembler used by C68.

New file transfer programs let you transfer files between QL and Amiga disk format. Two Amigados icons use the QLDIR program to show the directory of a QL disk in an Amigados window. File names, sizes and dates are shown.

Other Amiga CLI commands will copy or format Qdos disks, and transfer individual files in either direction. These commands were programmed in C by Mark Swift and his brother Francis. They build upon the Public Domain 'Messydos' disk driver, which also lets you use PC and ST disks in Amiga drives.

Verdict

If you are familiar with the QL, and already own an Amiga, you should get the Qdos Emulator. Version 3.10

works well, despite sluggish disks, but version 3.20 should be better still and it is worth looking out for the upgrade, scheduled for release later this year.

If you are only interested in a faster version of Qdos, Miracle's Gold Card may represent a better upgrade for

seems sure to swell the ranks of Qdos enthusiasts.

PD suppliers

The Amiga Qdos emulator is widely available from PD libraries, but many offer an incomplete or obsolete

package. Version 3.03 expects a German Amiga keyboard, unless patched, and corrupts QL disks so that only it can read them!

The full version 3.10 includes both German and UK versions, over 50 A4 pages of documentation in English, and another 30 pages in German detailing two public-domain toolkit files. 3.10B includes the first release of programs to transfer files between Qdos and Amigados, updated to version 2 for release 3.20.

Version 3.10 comes with full GST assembler source, and a ramdisk driver, on a separate QL disk. The source for version 3.20 is on an Amiga disk, with the assembler program.

Also available is a disk of utilities in QL format, including the TAS replacer, a

QL enthusiasts; it supports Mode8, bigger disks, and has no problem with TAS instructions.

If you are already attracted by the hardware and software of the Amiga, or find the QL hardware constricting, the Amiga Qdos emulator is an attractive proposition - it gives you the best of both worlds, letting you retain your investment in QL software, and explore the commercial and PD packages for the Amiga.

Rainer Kowallik and Mark J Swift have done a good job of implementing Qdos on the Amiga; the result merges the best of the Amiga - the hardware with the best of the QL - the software - in a combination that will please Motorola fans everywhere, and

patched copy of Quill, founts and other odds and ends. You should be able to get the latest set from any of these suppliers:

CGH Services, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA. CGH can also supply an extensive range of Qdos PD, including Mark J Swift's prize-winning 'Head Turn' animation.

QubbeSoft, 38 Brunwin Road, Rayne, Braintree, Essex CM7 5BU. QL PD specialists, they supply only the UK keyboard version in their Amiga Qdos 3.10 package.

Softville Computer Supplies, Unit 5, Stratfield Park, Elettra Avenue, Waterlooville, Hants, PO7 7XN. Softville specialise in PD for Amiga and Atari ST.

```
100 REMARK INSTALL code in Amiga chip RAM
110 REMARK by MJ Swift & SN Goodwin, 8-92
120 REMARK Uses FLEN (Toolkit 2)
130 :
140 FIND_CHIP
150 base=sv_chnop
160 PRINT space;" bytes available"
170 INSTALL "fip1_diy_code"
180 INSTALL "fip1_speedscreen_code"
190 INSTALL "fip1_turbo_toolkit_code"
200 PRINT space;" bytes remaining"
210 :
220 DEFINE PROCEDURE FIND_CHIP
230 addr=49152
240 REPEAT loop
250   nxt=PEEK_L(addr)
260   IF NOT nxt THEN EXIT loop
270   addr=nxt
280 END REPEAT loop
290 sv_free=PEEK_L(SYSBASE+12)
300 sv_chnop=PEEK_L(SYSBASE+124)
310 IF addr>sv_free
320   PRINT "ROMs loaded high in memory"
330   IF sv_chnop=sv_free
340     PRINT "No free chip memory"
350     space=0
360   ELSE
370     PRINT sv_chnop;" to ";sv_free;" free"
380     space=sv_free-sv_chnop
390   END IF
400 ELSE
410   PRINT sv_chnop;" to ";addr;" free"
420   space=addr-sv_chnop
430 END IF
440 END DEFINE FIND_CHIP
450 :
460 DEFINE PROCEDURE INSTALL(file%)
470 size=FLEN(file%)
480 IF size>space
490   PRINT "No room for ";file%
500 ELSE
510   LBYTES file%,base
520   CALL base
530   base=base+size
540   space=space-size
550 END IF
560 END DEFINE INSTALL
570 :
580 DEFINE FUNCTION SYSBASE
590 RETURN 16384:REMARK Default
600 END DEFINE SYSBASE
```


HERMES IPC CONTROLLER

The Hermes chip replaces the QL's intelligent peripheral controller. Ian Bruntlett tries it out.

Hermes is a replacement for the QL's 8049 Intelligent Peripheral Controller chip. The QL's 8049 had bugs, as did other replacement 8049s developed for QL add-on keyboards. With this in mind, Minerva author Laurence Reeves has rewritten the program in the QL's 8049, and the result is Hermes.

Installing a Hermes is as straightforward as installing Minerva - take the lid off, lever out the old stuff, put the new stuff in its place and put the lid back on. With one exception: after I had installed Hermes, removing the old 8049 with difficulty, I was told that the 8049 is probably the most awkward QL chip to remove. A thin screwdriver and plenty of patience is needed. Hermes has run on unexpanded QLs, Trump Card QLs and Gold Card QLs without problems.

The 8049 only handles serial input, so if you are having trouble sending data to another device, Hermes won't cure it. However, it is possible to drive a modem directly from the QL's serial ports now without having a special device such as Qconnect. Serial input has been enhanced so that input at 19200 baud now works, and input may be received with just one stop bit. The serial ports are usually restricted to a single baud rate. With Hermes it is possible to set the baud rates used for input. I have no devices handy to try this out.

Keyboard

The keyboard behaves better with Hermes installed. On a normal QL keyboard, the user subconsciously compensates for the roll-over. This is only noticed when moving from a Hermes machine to a normal machine and back again. When you start typing on a Hermes-driven keyboard, eventually you begin to take the better keyboard-handling for granted. Moving back to a normal QL keyboard results in occasional typing errors and wondering about the relatively poor keyboard handling. As

well as improving key roll-over, Hermes produces a quiet click on the QL's loudspeaker every time a key is pressed, giving instant feedback on whether or not a keypress has been properly registered. The key click can be switched on or off from software or from a key-press, Ctrl-Alt-Esc-4.

If you were hoping that Hermes will provide a safe non-maskable interrupt (NMI) via Ctrl-Alt-7, you will be disappointed. The advertising literature states that it is done cleanly, but in practise I found that there is little difference between them. On a QL without software to handle a NMI, generating a NMI invariably freezes your machine, requiring a hardware reset. On a QL with software to handle an NMI, generating an NMI will either cause the special NMI routine to be run, or crash your QL (this occasionally happens). The only difference between Hermes and a normal machine is that Hermes will put a few junk characters into the keyboard queue. As far as I am aware, Ctrl-Alt-7 (changed to Ctrl-Alt-Esc-7 on Hermes) is handled more cleanly on a normal machine. The extra characters generated by Hermes proved to be very inconvenient.

Sound

The 8049 handles sound, and Hermes improves the sound generation. When Sysmon started wailing, it sounded slightly better. But the QL's sound is ignored by so many programs that I think this improvement will rarely be used. Hermes expands the commands that may be sent to the 8049 and a toolkit is supplied to allow easy access from SuperBasic. Hermes loses some of its shine here. The boot file provided for the toolkit doesn't allocate enough space, resulting in SuperBasic getting stoned. If you have Toolkit 2, LRESPR flp1_ipcexts_bin will load the toolkit properly.

The extra Hermes commands are performed by extended "microdrive reduced

INFORMATION

Product: Hermes
version 2.18
Supplier: Tony
Firshmann Services,
12 Bouverie Place,
London W2 1RB.
Answering machine
071 724 9053.
Price: £25.
Contents: 8-page
A5 manual, utility
disk, replacement
8049 chip. To
upgrade from
Minerva Mk I to
Minerva Mk II, sent
to TF Services: All of
Minerva Mk I: the
manual, Minerva
eprom and board,
plus a blank disk
and £35.

sensitivity" (MDRS) calls via the MT.IPCOM / SMS.HDOP trap. The information in the Hermes manual assumes so much familiarity with MDRS that if you haven't used it before, tracing ipcexts_bin with Qmon seems to be the best way to find out what is going on. This is unsatisfactory - at least a line of hex could have been provided for each new MDRS command, to show where the parameters are filled in.

Despite doing all the jobs of the original 8049 and more, some of the connections to the Hermes are now spare (pseudo I/O via P20, P23, P26; I/P via T0, T1). I can't help thinking that a Hermes that could handle an IBM keyboard directly would sell like hotcakes. But there probably isn't enough room inside Hermes to have code for both the QL keyboard and an IBM keyboard.

Hermes is only really useful to people genuinely aggravated by the old 8049's failures. If you have a problem that it fixes, it is manna from heaven; otherwise, it's a technical curiosity.

Although it may seem like a contradiction, the place to start when designing a page on a desktop publishing (dtp) program is at the end. In other words, the way in which you are going to reproduce your document is going to influence many of your page-design decisions, such as number and width of columns; the size, style and boldness of fonts; and the form that illustrations will take. I suspect that most users of QL desktop publishing programs will be reproducing their document by photocopying, which raises special problems which I shall be referring to in this article.

One of the essentials of page design, no matter what form of reproduction you use, is to plan the page as a grid. Newspapers and magazines are produced with a number of columns to each page, and these columns are usually repeated throughout the publication, although more than one arrangement or number of columns may be used (as in *QL World*, which uses mainly three-column and four-column page formats). There is also a division of the page vertically by the use of column breaks. One of the strengths of *Professional Publisher* is that the grid is displayed on the screen with a default value of four columns and three breaks, dividing the page into 16 sectors. The default settings can easily be changed. *Front Page Extra* is less flexible, as no grid is shown on the screen, but from one to four columns can be used when importing text files. The weakest program in this respect is *Desktop Publisher*, which displays neither a grid on the screen, nor encourages you to think in columns and breaks when importing text.

Column numbers

Although you should maintain a consistent design throughout your publication, this does not mean that you have to stick to the same number of columns on each page. If you are using five columns per page you could design a page of two columns, one large and one small. If the smaller column is the width of two of the five columns, and the larger three of the five columns, the overall consistency of the publication is maintained. Newspapers and magazines often use columns in this way to separate news and features from editorial comment. Even if you are designing an advertisement, you should still think in terms of columns. This helps in such decisions as: where to place parts of the advertisement that need to be emphasised, such as special offers, the company logo, and the address and telephone number of the company. If you look through the advertisements in this edition of *QL World*, you should see how various companies have applied this principle.

The grid is so important that it is sensible to make a template of it to build up your pages. With *Professional Publisher* this is an easy, if tedious, process. Set up the grid that you want on the screen, then go to Mode Draw and use suitable brushes to mark the corners of the grid. I find the four 'corner' brushes best for this purpose. Finally, make a hardcopy of the page. If you possess *Front Page Extra*, import a full page of text in the column layout that you wish to use and print this out. Using the screen scroll commands (Shift and cursor)

you can note where each screen begins and ends on your hardcopy, and use these as column breaks.

You may have difficulty in printing out a whole page. If this is so, consult your printer manual to see if there is a way of cancelling the paper-end detector. This could be done by changing one of the dip switches, or by sending control commands to the printer. I have an Epson-compatible printer, and can switch off the page-end detection by one line of Basic:

```
OPEN #3, ser1 : print
#3, CHR$(27);"8"; :
CLOSE #3.
```

You will find your grid template very helpful when adding illustrations. If you don't own a digitiser or a scanner, you will have to add illustrations after printing out the document, using traditional paste-up. The grid template will show you how much space should be reserved for the illustration. In general, illustrations should fit an exact number of columns. Otherwise, there are a number of possibilities: the illustration can be cropped to the right size, it can be placed in a box, or a caption can be placed to the side of it. *Professional Publisher* also allows the text to flow round the illustration, and has routines for increasing and decreasing the size of the image.

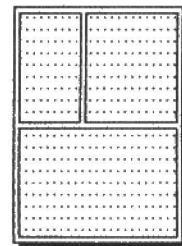
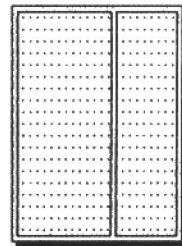
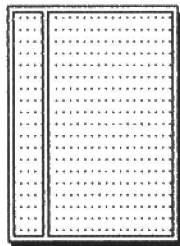
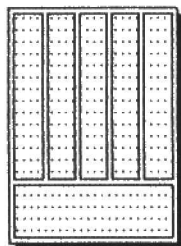
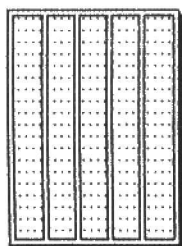
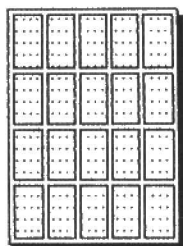
Easy to read

Adding the text to the page may seem easy, but it can be quite complicated. We have all experienced a book or article which was interesting and well-written but, for some reason, difficult to read. What probably was wrong was that

A Question of Dots

Part 3

In the third of his articles on desktop publishing, Geoff Wicks describes how to start your own page design by beginning at the end.



A basic grid (far left) and various page designs consistent with the grid.

your eyes were having to work too hard because of the way the type was laid out. There are several points to bear in mind: the first essential is that the letters themselves are unambiguous. There are two sorts of font: 'serif' and 'sans-serif'. Serifs are the small horizontal flourishes at the tops and bottoms of letters. Serifs help the eyes recognise a letter, which can be more difficult with a sans-serif font, especially a very thin one. Type the word "ill" with a capital I in both serif and sans-serif fonts to see the main problem for yourself. The traditional advice is to use serif fonts for large amounts of text, as they are more easily read than sans-serif fonts, which in turn are often better for headlines.

However, sans-serif fonts if properly sized have a less cluttered appearance, which helps general readability, and are now used in many publications. If you are not using reduction techniques at the reproduction stage, you may prefer to use sans-serif fonts. The smaller fonts suitable for large quantities of text are produced on a small grid, and if a part of the grid is used for the serifs the resolution of the main body of

the letter is slightly reduced. Also, the serifs are relatively large in relation to the main body of the letter and this can spoil the appearance of some letters. All these are reasons why some fonts, both serif and sans-serif, are much more popular for text publication than others.

The boldness of the font is also important. For large quantities of text a bold font is unattractive and tiring on the eyes. This is especially so if you are photocopying your document. In general, it is better to use fairly fine fonts if your reproduction method is photocopying although in some fonts the bolder type can look more like a typeset font. It is also good practice not to judge the suitability of a font from the output of your printer, as this may be misleading. It is better to make a photocopy of a small sample to see what

the final effect will be like. Proofing is an essential part of compiling any publication.

Moving eyes

Font choice is relatively easy, but it is more difficult to make the right choice of the number of columns, column width and font size. Given the right combination, the eyes can move easily from line to line. If there are too few letters per column, the eyes have to move rapidly, as if watching a ping-pong competition at close range. If the column is too wide, the eyes get to the end of the line and then have to go back and search for

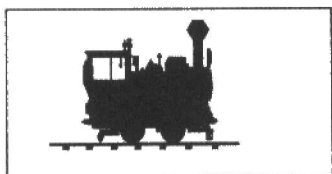
where the next line begins. One of the reasons why large quantities of typescript is tiring to read, is that the combination of letter size and column width is not good.

Desk top publishing helps you get it right.

For large quantities of text on an A4 page, I would advise a minimum of two columns and a maximum of four. One simple test to see if your columns are too narrow or your font too large is to try the effect of importing right-ranged text. If you have either excessive hyphenating or large white spaces in some lines, then you can be fairly certain that the ratio between font size and column width is poor. If you are a Professional Publisher user I would suggest that Small3 is the best font to use if you are not using reduction techniques at the printing stage. If you are reducing from A3 to A4, or from A4 to A5, you could try Helvetica. (In later versions of Professional Publisher this type of font is called 'Excellent'.) Finally, if you use reduced width and length printing there are a number of suitable fonts, with Times (or 'Roman' in later versions of Professional Publisher) and Baker being the most attractive although these may not photocopy satisfactorily. In that case, the enlarged version of Small3 can be used. Front Page Extra users do not have this choice, and will have to use CSIZE 1,0 for the best

You should not be afraid of using white space. This is particularly important if you have only one column in your document, when wide margins can help to improve the appearance.

Four ways of fitting a picture into a column:



You will find your grid template extremely helpful when adding illustr-

Use of a box



Steam Engine

You will find your grid template extremely helpful when adding illustr-

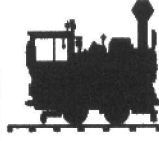
Use of caption

ations to your text. Not all users of desk top publishing own a digitiser or a scanner, and thus will often have to add illustrations after printing out the document using traditional cut and paste methods. Use



You will find your grid template extremely helpful when adding illustr-

Change of picture size



You will find your grid template extremely helpful when adding illustrations to your text. Not all users of desk top publishing own

Use of flowaround

ations to your text. Not all users of desk top publishing own a digitiser or a scanner, and thus will often have to add illustrations after printing out the document using traditional cut and paste methods. Use

a digitiser or a scanner, and thus will often have to add illustrations after printing out the document using traditional cut and paste methods. Use of the grid template will give a good idea how much space should

Adding interest to a page:

In addition to the legibility of individual lines, it is also important to pay attention to the general appearance of a page. A large grey, uniform block of text is hardly likely to encourage people to read your document.

Adding interest.

If you cannot break up the page with illustrations, then there are several ways in which you can add interest to the page. You should

In addition to the legibility of individual lines, it is also important to pay attention to the general appearance of a page. A large grey, uniform block of text is hardly likely to encourage people to read your document.

If you cannot break up the page with illustrations, then there are several ways in which you can add interest to the page. You should

In addition to the legibility of individual lines, it is also important to pay attention to the general appearance of a page. A large grey, uniform block of text is hardly likely to encourage people to read your document.

If you cannot break up the page with illustrations, then there are several ways in which you can add interest to the page. You should not be afraid of using

In addition to the legibility of individual lines, it is also important to pay attention to the general appearance of

Use plenty of white space

a page. A large grey, uniform block of text is hardly likely to encourage people to read your document.

If you cannot break up the page with illustr-

Use of paragraphs

Use of bold text

Use of dropcaps

Highlight from text

Illness Illness

A serif font (left) is less ambiguous than a sanserif font (right).

effect. You can however try different low resolution fonts and especially the 'fat' fonts.

White space

In addition to the legibility of individual lines, pay attention to the general appearance of a page. If you cannot break

up the page with illustrations, there are several ways in which you can add interest to the page. Whatever your number of columns, it is important to break up your text into paragraphs that are not too long, and every so often include a subheading. This should usually be in a bolder font than the main text.

High resolution fonts in Desk Top Publisher and Professional Publisher can be manipulated in a number of ways to produce among other things inverted, italic and bold versions, all of which can be useful if you want to highlight display text. Sometimes the difference between a normal font and its bold version is difficult to see, especially if the document is photocopied. In such cases

you should use a different font for subheadings or the italic version when highlighting sections within the body of the text. It may be worthwhile to use the font editor to make an extra bold version of the font. It is probably better to avoid the inverted versions of the fonts except for the larger

letter sizes. They do not reproduce well, especially when photocopied, and are wasteful on the printer ribbon.

There are a number of other ways of adding interest to a page. The first paragraph of each article

can be printed in a bold font. Drop caps can be used these are giant capital letters at the start of an article. Important sections of an article can be printed in a bold or italic font and a quotation from the article in a larger and bolder font than the text size can be printed on the page. Finally, Professional Publisher users can also lighten a page that

looks too 'grey' by increasing the space between letters when importing text. The default is one space, but I have found that a space of two or three (depending on the font) gives a more readable text. I have reconfigured my version of Professional Publisher to a default of three spaces.

Upside-down Greek

Sometimes it is difficult to judge the quality of the design of a page because you are too familiar with the content. Commercial publishers get round this problem by setting up a similar design with a meaningless text. This is called 'Greeking' - although the text used is usually Latin. Another suggestion I have seen is to view the page upside down, or from the back by holding it up to a light or a mirror. If the design is good, it will still look good the wrong way round!

If you are a *Front Page* or *Desk Top Publisher* user, not all techniques can be used. However, *Professional Publisher* users who are prepared to look around and to take the time to experiment, will discover possibilities in the program that they did not realise existed. In my next article I will discuss some of these, and also look at the use of clip art.

A Question of Dots

FONT EDITOR

Altering the QL font is easy if you know how to do it. Rich Mellor presents a method.

Recently there has been correspondence in Open Channel concerning POKES to alter the QL's fonts. A SuperBasic program was submitted, which, although it worked on a standard QL (and only channel #0 at that), may not work on all QLs. Such routines are only ever likely to work on unexpanded QLs, and will most definitely not work in a program running under the Pointer Environment, which stores additional information about each window. It was therefore decided to provide the following machine code solution, which is in the Public Domain and may be freely used in any program.

The origins of fonts date back to the earliest forms of printing. A printer would have

one standard set of letters and symbols which he used to produce documents. The more wealthy printers would have more than one set of characters, to enable them to embellish documents. Each of these sets of characters was known as a font (or fount), and this term is still used throughout the print business. Its basic meaning is 'a set of letters, numbers and symbols all designed in one particular style and size'. This distinguishes it from a typeface, which is a set of letters, numbers and characters all of the same design, regardless of size.

Two fonts

On the QL, each window can have two fonts attached, the first of which defines the characters from 0 to 127 and the second of which defines the characters from 127 to 255 (the reason that CHR\$(127) is defined twice is that the definition in the first font is the definition used for the copyright symbol, whereas the definition in the second font is that used for an unprintable symbol). Normally this is not noticeable and when you open a new screen channel (scr_ or con_), the standard QL fonts are attached to that channel (Minerva users can actually now use the command POKE_L !124!40,font0 and POKE_L !124!44,font1 to alter these default fonts and thereby attach a user-defined font to every single window that is opened after this command).

On a QL, a font is stored in the following format:

Offset	Value
\$0	Byte giving first character defined in the font
\$1	Byte giving the number of characters defined (minus 1)

\$2 ... \$A Nine bytes defining the first character
\$B ... Nine bytes each character....

Each character contained in the font is designed on a grid of 8 pixels wide by 9 pixels high, and therefore the easiest way of calculating the nine bytes which make up each character is to design the character on a sheet of graph paper (if you do not have a font design program). Then each row must be converted from binary to decimal. For instance, this is the binary representation of the character 'k':

```

BIT 7 6 5 4 3 2 1 0
0 0 1 0 0 0 0 0 0 = 64

```

```

1 0 1 0 0 0 0 0 0 = 64
2 0 1 0 0 0 1 0 0 = 68

```

```

R 3 0 1 0 0 1 0 0 0 = 72
O 4 0 1 1 1 0 0 0 0 = 112
W 5 0 1 0 0 1 0 0 0 = 72
6 0 1 0 0 0 1 0 0 = 68
7 0 0 0 0 0 0 0 0 = 0
8 0 0 0 0 0 0 0 0 = 0

```

Therefore a small program using the supplied toolkit routine to set the character 'a' to the same as the character 'k' in channel #1 would be:

```

10 a=RESPR(11)
20 POKE a,97 : REMark
97=CODE('a') - first character in font
30 POKE a+1,2: REMark
Only one character is being redefined

```

Listing One

* Public Domain Font Initialisation Routine.
* S_FONT V1.02 Rich Mellor

```

BV_CHBAS equ $30
BV_CHP equ $34
BV_RIP equ $58
IOW_FONT equ $25
SB_INIPR equ $110
SB_GTINT equ $112
SB_GTLIN equ $118$$
begin lea.l define,a1
      move.w SB_INIPR,a2
      jmp (a2)$$

define dc.w 1
      dc.w start-'
      dc.b 6,'S_FONT',0
      dc.w 0
      dc.w 0
      dc.w 0

start cmpa.l A5,A3
      beq.s no_param
      tst.b $1(A3,A6.L)
      bmi.s get_chan

no_param moveq #1,D1
      bra.s get_start

get_chan move.l A5,D5
      lea.l 8(A3),A5
      move.w SB_GTINT,A2
      jsr (A2)
      tst.l D0
      beq.s got_int
      bad_par moveq #1,D0
      rts

got_int move.l A5,A3
      move.l D5,A5
      move.w $0(A1,A6.L),D1
      addq.l #2,A1
      move.l A1,BV_RIP(A6)

get_start mulu #28,D1
      add.l BV_CHBAS(A6),D1
      cmp.l BV_CHP(A6),D1
      bge.s err_no
      move.l $0(A6,D1.L),D0
      bmi.s err_no
      move.l D0,A0
      move.w SB_GTLIN,A2
      jsr (A2)
      tst.l D0
      bnc.s bad_par
      subq.w #1,D3
      beq.s font2
      subq.w #1,D3
      bne.s bad_par
      move.l $4(A1,A6.L),A2
      move.l $0(A1,A6.L),A1
      moveq #IOW_FONT,D0
      moveq #-1,D3
      trap #3
      tst.l D0
      rts

err_no moveq #-6,D0
      rts
END

```

Link in new Basic command then return to Basic

One procedure - begins at 'start' - is called 'S_FONT' End of procedures No functions End of functions

Fetch channel number (if any) None, so use default channel (#1) Is there a hash symbol? Yes, so find channel.

Default channel = #1

Store parameter number.

Fetch one integer channel number.

Any error? Okay. Bad Parameter error.

Move onto next parameter

Get Basic channel number and convert to Qdos internal channel ID Reset BV_RIP

Point to Basic channel table.

Is it within the table? - No Is channel open? - No Store channel number Fetch the base of new font. - This is supplied as long integer

Oops. Only one font to be set? Yes, then set it No more than two fonts can be set! Oops - too many fonts. Store second font address. Store first font address.

This Job. Set the new font address. An error?

Signify channel not open

```

40  RESTORE
50  FOR i=0 TO 8:READ
bit:POKE a+2+i,bit
60  DATA
64,64,68,72,112,72,68,0,0
70  S_FONT #1,a

```

When designing fonts for use on the QL, it is however important that certain rules are followed to ensure that when characters are printed on screen, they have the desired appearance in all screen modes. The QL converts a given character into the desired mode by doubling up various rows or columns (depending on the size set using the CSIZE command). If you intend to redesign letters, the rightmost two columns (bits 0 and 1) should always be left empty, as these are ignored in current versions of the QL rom.

Old roms

The leftmost column (bit 7) is normally left blank to ensure a gap between adjacent characters, however, should you want to use this column as well, you must be wary of the fact that pre-JS roms cannot display correctly fonts which use this column (bit 7) and so this should always be left blank unless you know that your font is going to be used specifically on later roms.

You should also be aware of the fact that an extra blank row is generally left between each line of characters. Toolkit II users can however prevent this by using the command CHAR_INC.

Public domain

There are various sets of fonts available from the public domain, and sets are also supplied with the Turbo compiler, Speedscreen and Lightning; all of which can be safely used with this routine.

Having designed your new font, how do you get the QL to accept it? This is where the machine code procedure S_FONT comes into play. To add the new command to Basic, you will need to either RUN the standard hex loader program given in Listing two or enter the assembly code given in Listing one into your

own assembler (this was originally written using the MetaComCo assembler). Once the code file has been generated, load it using LRESPR from Toolkit II or this sequence of QL commands:

```

x=RESPR(140)
LBYTES flp1_FONT_ext,x
CALL x

```

You may wish to change the device or file name in the second line to suit your own system.

The machine code starts by linking in one new command, S_FONT, into the Basic name table. Once this has been done (for example with the CALL command listed above), then control is returned to SuperBasic. When the command is called it first of all tries to grab the channel number of the window where the font is to be attached. If no channel number is supplied, or if the parameter is not preceded by a hash (#) symbol, then it assumes the default channel (#1) is to be used. The command then checks whether the given channel number is valid and if so, then it grabs the address for the new font (or fonts). A 'Bad Parameter' error will be returned if no address is given for the start of the font. Having fetched these two addresses, it then calls the machine code routine IOW.FONT (in SMS-2 notation - also known as SD.FOUNT) to attach the given font(s) to the desired channel.

Font format

The syntax for the command is S_FONT [#channel,] font1 [,font2]. This command therefore allows you to set both fonts for any specified channel easily and quickly. Problems may occur if the fonts are not in the correct format (see above). The fonts can be reset to their original status by using S_FONT #channel,0,0. This command has the advantage that programs written using this command will work on all Qdos implementations (and SMS2 implementations), and even under the Pointer environment without alteration.

Listing Two

```

100 REMark QL World Hex Loader v 3
110 REMark by M Jeffrey & S Goodwin
120 :
150 CLS:RESTORE : READ space: star
t=RESPR(space)
160 PRINT 'Loading Hex...' : HEX_L
OAD start
170 INPUT 'Save to file...';f$
180 SBYTES f$,start,byte : STOP
190 :
200 DEFine FuNction DECIMAL(x)
210 RETurn CODE(h$(x))-48-7*(h$(
x)>'9')
220 END DEFine DECIMAL
230 :
240 DEFine PROCedure HEX_LOAD(start)
290 byte = 0 : checksum = 0
300 REPEAT load_hex_digits
310 READ h$
320 IF h$='*' : EXIT load_hex_di
gits
330 IF LEN(h$) MOD 2
340 PRINT'Odd number of hex
digits in: ';h$
350 STOP
360 END IF
370 FOR b = 1 TO LEN(h$) STEP 2
380 hb = DECIMAL(b) : lb = DEC
IMAL(b+1)
390 IF hb<0 OR hb>15 OR lb<0
OR lb>15
400 PRINT'Illegal hex digit
in: ';h$ : STOP
420 END IF
430 POKE start+byte,16*hb+lb
440 checksum = checksum
+ 16*hb + lb
450 byte = byte + 1
460 END FOR b
470 END REPEAT load_hex_digits
480 READ check
490 IF check <> checksum
500 PRINT'Checksum incorrect.
Recheck data.':STOP
520 END IF
530 PRINT'Checksum correct, data
entered at: ';start
560 END DEFine HEX_LOAD
570 :
580 REMark Space requirements
for the machine code
590 DATA 140
600 :
610 REMark Machine code data
620 DATA '43FA0008347801104ED2'
630 DATA '0001001006535F464F4E'
640 DATA '5400000000000000B7CD'
650 DATA '67064A33E8016B047201'
660 DATA '60222A0D4BEB00083478'
670 DATA '01124E924A80670470F1'
680 DATA '4E75264D2A453231E800'
690 DATA '54892D490058C2FC0028'
700 DATA 'D2AE0030B2AE00346C2C'
710 DATA '203618006B2620403478'
720 DATA '01184E924A8066CE5343'
730 DATA '6708534366C62471E804'
740 DATA '2271E800702576FF4E43'
750 DATA '4A804E7570FA4E750000','*',11006

```


In this article, I should like to look at some SuperBasic programs that can make the life of the Psion suite user a little easier. From time to time I have found myself writing short routines to do things that either the Psion programs cannot do, or do in such an awkward way that I have felt the need to look for a better. The listings that accompany this article are tidied up versions of these but as usual they are not intended to be the be-all-and-end-all, but just to demonstrate the sort of thing that can be done: modify them to suit your own needs.

They all involve OPENing files directly on disk or other storage medium without actually getting them into the computer in the usual way (with LOAD, EXEC etc). I remember that the first time I did this it was with some trepidation, since I felt that if something went wrong I might lose the file altogether; but rest assured, all but one use OPEN_IN on existing files, which enables you get information out of the file (read it) but not to alter it (write to it). Anyway, you always ought to have a back-up on a different disk/cartridge just in case. I generally do all this sort of manipulation on ramdisk files since the access time is much shorter even than disk and the programs are quite slow in operation. If it works, then I copy the results onto a permanent storage medium.

Configuring

Of course there is a good precedent for this approach, some members of the Psion suite itself are written in SuperBasic, namely INSTALL_BAS and CONFIG_BAS, and I'd like to start with them. When Psion designed their suite of programs all those years ago,

all they had for them to run on was an unfinished, unadorned 128K QL with microdrives. Luckily someone had the foresight to realise that all kinds of wonderful things might be possible one day and they included a utility program called "config_bas". This enables users to reconfigure Quill, Abacus, Archive and Easel to run from devices that were scarcely even a twinkle in Miracle's eye at that time. In practice what it does is to enable you to change references within the code of Quill, etc. from "mdv1_", "mdv2_" to "flp1_", "ram8_" etc as required.

What Psion failed to think of was the need to provide a similar facility to work on config_bas and its companion, install_bas, themselves, with the result that you may find yourself using them on microdrives and copying the result to your favorite device later. You can buy programs to read through a SuperBasic program and effortlessly and mindlessly convert all references to "mdv" to "flp", and you can find them lurking about in user group libraries too, but all readers may not have access to those, so here

is a set of DIY instructions.

One way around this is to use the FLP_USE MDV command. This is available in everyone's disk interface toolkit, but I have never felt this to be very satisfactory, since it's confusing and prevents you from using the real microdrives while it is active. A better solution is to edit the programs themselves; all you have to do is to run through the listing picking out all the lines that contain the offending words "mdv", "microdrive" and "cartridge" and replace them with "flp", "diskdrive" and "disk". Or, you could but, why not get the QL to do it for you. After all one of the reasons you bought the darned thing was because it could carry out repetitive boring tasks without complaining, and faster and more reliably than you could do it yourself. (Or was it just to play Space Invaders?)

QL's brains

The program in **Listing one** only does part of the job, I've left something for me (or you) to do since I thought the task of deciding which occurrences of the strings

Listing 1

```
5 REMark
10 nm$="String_fisher"
15 REMark (pd) hjc 1992.06.19 ver 1.31
20 REMark ~~~~~Main
25 Setup: Fish: Finish: STOP
30 REMark ~~~~~
100 DEFine PROCedure Setup
105 LOcal i,src$, out$: CLS: CLS#0
110 ch% = 3: REMark ch% &ch%+1 unused channels
115 INPUT"Device and filename to search:"\,src$
120 INPUT"Device and filename for output:"\,out$
125 OPEN_IN#ch%,src$: OPEN_NEW#ch%+1,out$
130 RESTORE : READ num: DIM str$(num,20)
135 FOR i = 1 TO num: READ str$(i)
140 END DEFine : REMark ~~~~~DATA
145 REMark Number of strings + actual strings <<<<
150 DATA 3,"mdv","microdrive","cartridge"
155 REMark ~~~~~
200 DEFine PROCedure Fish
205 LOcal a$,i,loop,s$: REPEAT loop
210 IF EOF(#3): EXIT loop
215 s% = 0
220 INPUT#3,a$: AT#0,0,0: PRINT#0,a$(1TO 4)
225 FOR i = 1 TO num: s% = s%+str$(i) INSTR a$
230 IF s%: PRINT a$: PRINT#4,a$
235 END REPEAT loop
240 END DEFine : REMark ~~~~~
300 DEFine PROCedure Finish
305 CLOSE#3: CLOSE#4
310 END DEFine : REMark ~~~~~
```

In the first of three short articles, Howard Clase presents some types of SuperBasic program which make the Life of Psion a little easier.

actually needed replacing required more intelligence than the QL is capable of - it can be persuaded to change all of them, but not just the essential ones. What it does is to read through the file that contains the SuperBasic listing and fish out the lines which contain one or more of the target strings, printing them to a new file and to the screen as it goes along. It also prints the current line number to the command window, just so that you can see that something is happening (I hate programs that simulate a screen lockup for long periods!).

You are asked for a couple of file names (115,120) and since you must assign a channel number when OPENing a file you must ensure that the variable ch% (110) is given the value of the first of a pair of unused channels - if you already have your printer as channel#3, then change ch%. The offending words are read in (130) from DATA statements (145). You can modify the program to fish out any words you like from any Ascii file, just put them in at line 145 (and following if necessary), remembering to start with the total number of words since that is used in the loops at lines 135 and 225.

The business is done in a loop in the PROCedure Fish (205-240). Since you do not usually know the number of lines in a file in advance it's best to use a REPEAT loop for this sort of thing and EXIT when you get to the end of the file. I like to put the EXIT line first since you just might come across an empty file. The lines are INPUT one at a time into a\$ (220), and if it contains one of the target strings the flag, s%, is given a non-zero value (225), which Qdos interprets as "TRUE" (230), printing the line to the screen and into your "catch" file.

Finishing

Finally the files are CLOSED (305). If the program stops in midstream for any reason the files will be left open. The easiest way to

deal with that is to enter "Finish" at the keyboard; this applies to all the listings in these articles.

There are two ways to proceed from here. I LOAD the catch file, edit any references that look like actual devices, SAVE the altered lines as a temporary file, LOAD the original full version and then MERGE the temporary file on top of it. This will replace the original lines with the edited ones. If you do this don't try to RUN the collection of miscellaneous lines and don't RENUMber it. The alternative is to print out the catch file on your printer, and use this as a guide when editing the complete listing. The advantage of this approach is that you can see the lines in context, and it may help you to decide whether a particular change needs to be made or not. If in doubt, make it; even if it's a variable name it won't do any harm, as long as you change all occurrences of course. If you decide the second method is the one for you, you can type "ser1hc" (or whatever your printer's fancy is) when asked for the output file and it will go straight to your printer. (Qdos doesn't seem to mind if you OPEN_NEW a serial port.)

Having re-configured your files you can now compile them. Both will compile without problems with Q_Liberator, and the extra speed makes them a joy to use, especially with a Gold Card and Minerva. I do not have copies of any other compilers, so I cannot test them. I'd be interested to hear from any reader who has experience with either of DP's programs. If you use a compiled version you can ignore the message at the end of config_obj about resetting the machine. Unlike the interpreted version the compiled program releases all the memory it has set aside when you exit.

In the next part of this article, we'll be looking at importing text from other sources.

JOCHEN MERZ SOFTWARE

Im Stillen Winkel 12 - D-4100 Duisburg 11
Germany - Telephone & Fax: 0203 501274

It is done: QVME - the QL-Emulator for the Mega STE exists!

No soldering and unbelievable features: fully programmable, screen resolution, horizontal & vertical frequency etc. can be changed while the system is running! If used with a good Multisync monitor, you can get resolutions up to 1024x768 pixels!!! Or how about 896x896, or ... it is up to you! Of course, you can get the good old 512x256. It runs right now on the STE, we're working on the TT software and an adapter to put it into Mega STs. The price is **£256** - write for more details!

QD 4 - Editor using the Pointer Environment, with many new options: GOTO BASIC Procedure/Function, Label, improved parameter string, print, search/replace and MenuConfig (the same as Config, but menu-driven, new Vers.). **£38**. Upgrade from V3 **£10**

QSpread - Spreadsheet for the Pointer Env. More than 32000 cells. Completely pointer-driven, up to 3 horizontal and vertical splits at the same time. Formulae-orientated, 22 scientific functions, macro functions, help, scrap, etc. **£49**

QMenu - Menu Extension V3 QMenu is an interface to pre-defined menus (e.g. file-select, simple-choice boxes, select from lists) which are very easy to use. These menus may be used by SuperBASIC, machine code and other languages. The menus are very comfortable, multi-column etc. **£12.90**

Update from previous version with new manual **£5**

FiFi - the FileFinder FiFi is extremely useful and saves a lot of time: it scans devices or directory trees and searches files or filenames for strings. Combinations are possible, e.g. String1 AND string2 BUT NOT string3 etc. **£13**

QDOS Reference Manual - This book is a must for all machine-code programmers. It explains how to use QDOS, all traps and vectors, the Thing System, the HOTKEY Syst.II and much more. It shows which features work on a QL, an Emulator, how to write compatible for future operating systems. 170 pages. **£27**

QPTR - The Pointer Environment Toolkit - Revised manual which describes how to use the Pointer Interface and the Window Manager from SuperBASIC and machine code. Examples on disc, keys, macros & extensions for SuperBASIC. **£27**

FLP/ROM Level 2 with ATR!!! Replacement EPROM for QBoard (V1.17 onwards, with & without mouse) or Trumpcard (please specify!). Real sub-directories! About twice as fast, improved slave-block-handling (not on JM-ROMs) **£18**

EASYPTR II - Create your own Pointer-Env. menus & sprites, use them in your SuperBASIC or machine-code programs. Supports all the PE facilities, even split application sub-windows and uses the Menu Extension! Many examples! **£49**

SYSTEM - System Tools for QDOS: real file-attributes are added, e.g. write-only, hidden or for selected users only. Works even over net and on winchester. New TRAP 3. New devices (equiv. to QL-EMU): named pipes, MEM, NUL, 80 new procs & features, e.g. WSET, FATTR, WSET, FVERS, ADIR, ASTAT **£27.50**

QDesign II - more features, improved and more printer drivers (incl. laser), QSnap which scans text from screen and puts it into scrap, virtual drawing mode on Emulator and Minerva etc. **£48** Upgrade from version 1 **£13**

VecEdit Vector-Font-Editor for QDesign **£19**

NEW: SER Mouse V2 - software driver which lets you connect a serial (e.g. IBM-style) mouse to one of the SER ports of the QL, which then mimics the QIML file. On a 3-button mouse, you have ESC, Wake & Sleep in addition. **£14**

The Lonely Joker - three great patience card games. Needs Toolkit II **£14.90**

Diamonds £11 - BrainSmasher £12 - Arcanoid £10 - Firebirds £10 SuperGamesPack £25 - QShang £14 - QSUP System Utility Pack £26 Thing & Eprom Manager £18.50 - DISA (Intelligent Disassembler) £27

QL-Emulator Extended4 (as long as stock lasts) £139

Floppy adaptor to connect QL diskdrives to the ST **£9**

Very special offer: Harddisk unit for the ATARI ST (without harddisk): case for Syquest: changeable drive AND 5.25" or 3.5" drive, PSU, host-adaptor (100% QDOS compatible), cables, driver, thermo-controlled ventilator etc. **£120**

QDOS on a 68030 (in every Mega ST) running at 32MHz: incredible speed (approx. 8 times faster than a ST, 20 times a QL) together with new E-Driver-Software. Write for more details! Price: 32MHz Version **£600**, 25MHz **£500**

NEW: E-Driver software with new initialisation: boots faster, runs faster, needs less memory, boots from harddisk without EPROM, more features inbuilt etc. The way forward, as level D ends at D:16. With new manual **£17**

Technical Printer Manuals:

EPSON ECS-P/2 for all EPSON 9/24/48 pin printers. 500 pages **£27**

HP Deskjet for the Deskjet family (including 500C) **£15**

Please add **£4.50** for p & p (Europe) or **£4.50** for one item and **£2.50** for every further item (Overseas). All prices excl. V.A.T. E&OE.



Please write or call (English & German spoken) for more information. If you are interested in a Customer Support Mailbox, please write to us!

Perfect Paragraphs & Shining Characters

Simply driving his printer is not enough for Martin Harris; he teaches his printer to be perfectly proportioned. Here, he describes his aims and how he achieved them.

In this article, I am going to try and move beyond the concept of printer drivers, which others have written about adequately, in order to develop the full capacities of your printer. Even the simplest printer contains a wealth of possibilities largely ignored by its owner, either through word-processor inadequacy or, more likely, through fear of complexities. If you would like to produce pretty, professional documents instead of standard, right-justified, computerised-looking ones, then this article will interest you. There will also be quite a lot of discussion about programming, and I am indebted to Digital Precision for publishing some of the utilities referred to. I will deal with the matter in reasonably broad terms but, as my MT 85 printer uses the popular Epson protocols, there is a good chance that the resulting programs can be applied with very little change to your system.

Needs utilities

The program I have produced for my own use is reproduced in this part of the article for those to whom it may be useful. It is compiled with Turbo, and needs items from Digital Precision's Toolkit to run.

My "no frills" Mannesman Tally 85, 9-pin DMP (dot-matrix printer) has draft, enhanced and near-letter-quality type, enlarged, compressed, elite and pica character densities, bold,

italic, superscript and subscript fonts, proportional or non-proportional character widths, line spacing settable to n/216ths of an inch, underline, tab stops, a smattering of foreign characters, user definable draft quality characters and graphics. But most of the features of even this basic printer were never used.

My first word-processor, Quill, had a total of 14 translate codes - hopelessly inadequate. As I use a lot of foreign characters, I had six different printer drivers on separate tapes, each for a different purpose, and hoped (against hope!) that in any one document I didn't need capabilities that were not found on one and the same driver.

Free at last

Much of this changed with the advent of The Editor. With its unlimited number of "translates" and the freeing of the control characters for use as printer codes, I could have about 20 printer control codes and use all the possible foreign characters in the QL without it becoming at all messy. For the sake of completeness, my full printer driver (**Figure one**) is appended along with a few comments to give you the idea.

The code sequences are taken straight from the printer manual; e grave means 'switch to French character set (27, 82, 1), print e grave (125), switch back to USA character set (27, 82, 0)'; e circumflex is a little more complicated; the printer does not have this letter, but it does have the circumflex accent on its own. The codes mean 'switch to French character set, print e, backspace, print circumflex, switch to USA set'.

The toggles are meant to be treated like brackets; the first use of the code will generate the sequence up to the semi-colon and the second use of the same code will generate the sequence after the semi-colon and so on alternately. These codes simply mean switch on the facility then switch off the facility.

Problems and principles

So we now have a printer driver with which we may utilise all the features of the printer. Wonderful! So why am I writing this article? It turns out that there are many problems, which we must first try to define before proceeding further.

Problem 1

Due to what the printer manual calls a hierarchy, you cannot switch from elite to compressed type. You must first switch nominally back into pica, then into compressed; after switching out of compressed, an off-pica/on-elite toggle must be inserted correspondingly. This is only one problem arising from the printer's hierarchy, and there are all sorts of others in which the printer will not accept certain combinations of print modes. In such circumstances, the printer moves into some other mode which is consequently not monitored by the word-processor, resulting in havoc.

Problem 2

Proportional mode is not compatible with certain other options: (i) Compressed: total confusion results; (ii) Backspace: I need my printer to backspace because of certain accented foreign characters, as mentioned above. If proportional is on, the backspace simply doesn't work, so the accent appears on its own.

These are "mechanical" problems which could and should be solved by appropriate programming but there are still many others that consume much time because of checking, which we might call "human" errors.

Problem 3

For example, underline must be switched off at the end of every line and back on at the start of the next until the end of the underlined text is reached, otherwise the underline prints in the spaces that make up the margin. This is no problem for simple one-line headings but very awkward if several lines are involved and you decide

to reformat the paragraph. Actually, Quill sorts this problem out but, as we have decided to abandon Quill, we must find our own solution.

Problem 4

With toggles generally, there is always the possibility of forgetting to switch off the second toggle and ending up with the rest of the document in (for example) unwanted boldface. Digital Precision did warn us to be very careful with toggles, but must we be punished for the tiniest moment of fatigue? The error is worst when an end-underline has been missed - but it could apply to any other toggle. If, further, you enjoy a 15K printer buffer, the printer driver will already have processed several pages of document before you first notice the error on the page itself! The only solution I know is to reset the machine.

Problem 5

Density changes: the word-processor very naturally is ignorant of the density of characters in the printer. This gives rise to possible errors with, say, enlarged text exceeding the physical page width, or margins ending up in the wrong place in the case of changing density at the start of a paragraph. Proportional spacing is, of course, a special case of density changing with each character. This constitutes a problem in itself.

Problem 6

The most complex of all the problems is proportional spacing. I like this mode a lot. It obviates cheap-looking columnar text, and looks smooth and dignified. By contrast, full justification with non-proportional characters not only looks like a column, but also requires redundant spaces between words to pad out the lines; the end result is garish to my eyes. By analysing where to insert Ctrl-Ps, I find that I can produce professional looking, proportionally-spaced documents. It may not necessarily be of utmost importance to you to achieve this, but if you earn a living from your QL computer, bad

or cheap presentation of documents could mean the difference between a long and profitable relationship with a client, and a short one.

Alternatives

There are various alternatives. The one not to be considered at all is to opt for a DTP (desktop publishing) program; perhaps you have considered this, hoping that it would micro-justify very effectively. The operation is easy enough for simple text, but for heavily-formatted work requires an excessive amount of preparation. In any case, it will take twenty minutes per page to print if you only have a DMP, since the DTP program uses the graphics features of the printer.

Another alternative is to use a top of the range IBM compatible word-processor which uses a laser printer and a language called Postscript, which I understand is a graphics language. It seems that we can have decent-looking proportional

characters at the expense of right justification. But, as usual, the story doesn't end there! Consider the following text:

Screen:

(i) First thing heading
The cat sat on the mat
All that glitters is either
gold or not

Printed:

(i) First thing heading
The cat sat on the mat
All that glitters is either
gold or not

automatically with proportional mode on). Our revised text looks like this on the screen:

P(i) PFirst thing heading
The cat sat on the mat
All that glitters is either
gold or not

and prints like this:

(i) First thing heading
The cat sat on the mat
All that glitters is either
gold or not

Not quite WYSIWYG, but we are beginning to use the printer to produce a professional text. The text of the paragraph is proportional, but only the paragraph numbering is columnar.

(Technical note: we are trying to reproduce Martin's spacing as closely as possible on our professional DTP system - we may not be 100% successful. Fortunately he has described it very clearly. It does seem that DP's wordprocessor Perfection is a better shot at the WYSIWYG approach, and should assist

KEYING DISPLAY CODES SENT TO PRINTER

;Name Mannesman Tally 85 Euro			
;PDV, ser1			
;PBD, 9600			
;PHM, 4			
;PFM, 6			
;PPL, 60			
;PPC, 40			
;PPT, Page #r			
;PPS, ALTP			
;PXL e acute	ctrl shift 3	;é;	#27, #82, #1, #123, #27, #82, #0
;PXL e grave	ctrl 0	;è;	#27, #82, #1, #125, #27, #82, #0
;PXL e circumflex	ctrl 1	;ê;	#27, #82, #1, #101, #8, #94, #27, #82, #0
;PXL e trema	ctrl /	;ë;	#27, #82, #1, #101, #8, #126, #27, #82, #0
;PXL a grave	ctrl -	;à;	#27, #82, #1, #64, #27, #82, #0
;PXL a circumflex	ctrl .	;â;	#27, #82, #1, #97, #8, #94, #27, #82, #0
;PXL a trema	ctrl esc	;ä;	#27, #82, #2, #123, #27, #82, #0
;PXL i circumflex	ctrl 5	;î;	#27, #82, #1, #105, #8, #94, #27, #82, #0
;PXL i trema	ctrl 2	;ï;	#27, #82, #1, #105, #8, #126, #27, #82, #0
;PXL o circumflex	ctrl 8	;ô;	#27, #82, #1, #111, #8, #94, #27, #82, #0
;PXL o trema	ctrl shift \$	;ö;	#27, #82, #2, #124, #27, #82, #0
;PXL u grave	ctrl shift :	;ù;	#27, #82, #1, #124, #27, #82, #0
;PXL u circumflex	ctrl s/colon	;û;	#27, #82, #1, #117, #8, #94, #27, #82, #0
;PXL u trema	ctrl '	;ü;	#27, #82, #2, #125, #27, #82, #0
;PXL n wiggle	ctrl shift)	;ñ;	#27, #82, #2, #124, #27, #82, #0
;PXL Å trema	ctrl shift @	;Ä;	#27, #82, #7, #91, #27, #82, #0
;PXL Ö trema	ctrl shift D	;Ö;	#27, #82, #2, #92, #27, #82, #0
;PXL U trema	ctrl shift G	;Ü;	#27, #82, #2, #93, #27, #82, #0
;PXL Eszet	ctrl shift <	;ß;	#27, #82, #2, #126, #27, #82, #0
;PXL N wiggle	ctrl shift N	;Ñ;	#27, #82, #7, #92, #27, #82, #0
;PXL Spanish question	ctrl shift T	;¿;	#27, #82, #7, #93, #27, #82, #0
;PXL Spanish exclamat	ctrl shift S	!;	#27, #82, #7, #91, #27, #82, #0
;PXL English pound	£	;£;	#27, #82, #3, #35, #27, #82, #0
;PXL Para' indent	ctrl shift V	;§;	#27, #82, #1, #93, #27, #82, #0
;PXL Pesetas	ctrl shift P	;P;	#27, #82, #7, #35, #27, #82, #0
;PXL Yen	ctrl shift >	;¥;	#27, #82, #8, #92, #27, #82, #0
;PXL Per mille symb	ctrl shift Z	;‰;	#27, #82, #1, #91, #27, #82, #0
;PXL c cidilla	ctrl shift 9	;ç;	#27, #82, #1, #92, #27, #82, #0
;PXL trema backspace	shift f	;¨;	#27, #82, #1, #8, #126, #27, #82, #0
;PTG NLQ/Draft toggle	ctrl n	;N;	#27, #120, #1; #27, #120, #0
;PTG Bold/Norm. toggle	ctrl b	;B;	#27, #69; #27, #70
;PTG Enlarged toggle	ctrl g	;G;	#27, #87, #1; #27, #87, #0
;PTG Compressed toggle	ctrl h	;H;	#27, #71, #15; #18, #27, #72
;PTG ELite/Pica toggle	ctrl q	;Q;	#27, #77; #27, #80
;PTG Ppnl space toggle	ctrl p	;P;	#27, #112, #1; #27, #112, #0
;PTG Superscript toggle	ctrl s	;S;	#27, #83, #0; #27, #84
;PTG Subscript toggle	ctrl t	;T;	#27, #83, #1; #27, #84
;PTG Italics toggle	ctrl y	;Y;	#27, #52; #27, #53
;PTG U/line toggle	ctrl u	;U;	#27, #45, #1; #27, #45, #0
;PTG Line spacing toggle	ctrl	;D;	#27, #48; #27, #50

greatly in providing a more user-friendly screen. I see no reason why the system under discussion should not work with Perfection or any other wordprocessor, provided that it uses printer control codes of one character only, in conjunction with a printer driver.

The short of it ...

If you only ever type short documents of a few pages you are already on your way to producing top quality work. Your printer driver can operate every feature of your printer. You are aware of all the pitfalls and manually override them wherever necessary. You check that all your toggles are correct and adjust for printer hierarchy. You don't mind reformatting a paragraph even though it involves amending all the underlines. You make sure that, when you have a foreign character necessitating a backspace with overprint, proportional space mode is off. And so on according to your printer's own personality. Even if you make a mistake, it's not serious as you can reprint a document of two or three pages after making the correction.

But when you have a more complex and lengthy document and you cannot afford the two hours it takes to thoroughly check and correct the 30 pages or so, as well as insert the sometimes hundreds of proportional toggles, it begins to look as if a computer program might come in handy. Of course, it is essential to have done many times manually the operation you wish to computerise; firstly, so that you understand thoroughly what you are trying to achieve and secondly so that you prove the ultimate usefulness of the project as a time- and effort-saver.

We've looked at the types of problem in general, but will need to analyse them in more detail - but not before we have constructed a programming framework within which these errors can be solved.

What we need ...

The kind of program that we need is one which will take our typed and saved document, with its control codes, check that there are no errors, physically correct the most important ones, "correct" situations which we know well about but which are time consuming to do by hand, advise us of any probable errors so that we may decide to alter them or not, and finally to insert all the hundreds of proportional switches needed in a heavily formatted document with lots of sub-paragraphs, paragraph numbers, and so on. The program must then save the processed document back onto the medium, while preserving the original for reference.

We will need:

a) a screen to input the name of the document to be processed, and the name of the resulting document. As our program is basically a utility, the smallest screen is all that is required.

b) media checks

c) an overall program loop to examine each character in the original document and poke that character, plus any others that may be required, into a new document.

d) an error reporting system.

a) and b), (program lines 1080, 1320 to 1370, and 1540 to 1660 of Listing one), should be self evident; the media checks have been copied more or less from Digital Precision's own suggested uses of their toolkit which comes bundled with Turbocharge, the compiler which we will eventually use to compile our basic program. By the way, mine is a disk system with a 640K memory add-on and Toolkit 2, so you should have no trouble over compatibility with this simple configuration.

The program copies the original document into memory (lines 1400 to 1450) and eventually creates a second document in memory (lines 1700 to 1710) before saving it to disk (line 3270). This is extravagant if you only have the basic machine's memory, so some simple

```

1000 REM: Major_bas
1010 REM: Printer driver enhancement utility
1020 REM: Copyright Martin Harris 1992
1030 REM: Must be compiled with TURBO
1040 REM: Designed for use on Editor files with Epson
1050 REM: Protocol Printer
1060 REM:
1070 REM:
1080 OPEN #4, con_512x20a0x236_15
1090 IMPLICIT% n1,lines
1100 IMPLICIT% comp,precompel,comperr,bold,ital
1110 IMPLICIT% elite,sscript,offset,round,backstore
1120 IMPLICIT% frontstore,tab,lno,nitab,tally,a,b,c
1130 IMPLICIT% precomppnl,enlstart, enlarged
1135 IMPLICIT% eliterrstat,comperrstat,italerrstat
1137 IMPLICIT% bolderrstat,sscripterrstat
1140 IMPLICIT% better,bolderr,italerr,eliterr
1150 IMPLICIT% sscripterr,next_tabbed_line,tab_posn
1160 IMPLICIT% last_tab,ind_posn,letter,ppnl,uline
1170 IMPLICIT% ulinerr,indent,status1,status2
1180 IMPLICIT% status3,status4,status5,status6
1190 IMPLICIT% status7,succes,store,enlerr
1200 DIM info_array%(0,0):DIM error_array%(0,0)
1210 DIM errcount_array%(0):DIM errcount_array%(?)
1220 DIM error_array%(?,10)
1230 DIM er$(27):er$="ERRORS DETECTED AT LINES: "
1240 DIM sourcefile$(20):DIM destfile$(20):DIM tmp$(20)
1250 DIM key$(1):sourcefile$="flp2_":destfile$="flp2_"
1260 eliterr=1:comperr=2:enlerr=3:sscripterr=4
1270 bolderr=5:italerr=6:ulinerr=7
1280 REM: Having a main loop ensures that multiple
1290 REM: documents may be processed without having
1300 REM: to reexecute the program
1310 REPEAT main_loop
1320 REPEAT source
1330 PRINT #4, "Device / filename for proportional spac
e tidy: (x to quit) ";
1340 sourcefile$=EDIT$(#4, sourcefile$)
1350 IF sourcefile$="x":CLS #4:CLOSE #4:STOP
1360 IF DEVICE_STATUS(1, sourcefile$) < 0
1370 PRINT #4, "Error. Try again."
1380 ELSE EXIT source
1390 END IF
1400 END REPEAT source
1410 OPEN_IN #5, sourcefile$: SET_POSITION #5, 1.6E6
1420 length=POSITION(#5):CLOSE #5
1430 buffer1=ALLOCATION(length)
1440 IF buffer1<0:PRINT #4, "Insufficient memory.":NEXT
main_loop
1450 LBYTES(sourcefile$),buffer1
1460 lines = 1
1470 REM: Find out how many lines there are in the
1480 REM: source document since this will significantly
1490 REM: affect the size of the array "info_array"
1500 FOR lcount = buffer1 TO buffer1+length: IF PEEK(lc
ount)=10:lines=lines+1
1510 IF lines<2100:DATA_AREA 25
1520 IF lines>2099:DATA_AREA 70
1530 DIM info_array%(lines,4)
1540 REPEAT dest
1550 PRINT #4, "Enter device and filename for destinati
on file: ";
1560 destfile$=EDIT$(#4, destfile$)
1570 IF sourcefile$=destfile$
1580 PRINT #4, "Must be different device or name. T
ry again? ";
1590 IF YEA_OR_NAY
1600 NEXT dest
1610 ELSE CLS #4:CLOSE #4:STOP
1620 END IF
1630 END IF
1640 n=DEVICE_STATUS(2, destfile$)
1650 IF n>0 AND n<(length*1.05): PRINT #4, "Insufficien
t space on device.": NEXT dest
1660 IF n<0: PRINT #4, "Error. Try again.":NEXT dest
1670 EXIT dest
1680 END REPEAT dest
1690 n=DATE
1700 buffer2=ALLOCATION(length*1.05)
1710 IF buffer2<0:PRINT #4, "Insufficient memory.":NEXT
main_loop
1720 count=buffer2-1:lno=1
1730 total=(buffer1+length):number=buffer1-1
1740 REMark +
1750 REPEAT line_analysis
1760 ind_posn=0
1770 REPEAT start_of_line
1780 REM: info_array(lno,1) is filled with the posn of
1790 REM: the first genuine printable character
1800 number=number+1
1810 IF number=total:EXIT line_analysis
1820 letter=PEEK(number)
1830 SELECT ON letter
1840 -31 TO 32
1850 ind_posn=ind_posn+1:NEXT start_of_line
1860 -1 TO 9,11 TO 26
1870 NEXT start_of_line
1880 -10
1890 lno=lno+1
1900 NEXT line_analysis
1910 =REMAINDER
1920 number=number-1:info_array%(lno,1)=in
d_posn+1:EXIT start_of_line
1930 END SELECT
1940 END REPEAT start_of_line
1950 tab_posn=ind_posn:status1=0:status2=0
1960 status3=0:status4=0:status5=0
1970 REPEAT rest_of_line

```

changes will be necessary to read from and poke directly into the disk drives. This is not so bad, since disk access is buffered in the QL and most of our access will be sequential, reducing wear and tear on peripherals. However, doing this a lot will nevertheless stretch the drives to their limit; I also found that it took several times longer to process a document, so I opted for carrying out the whole task in memory.

Disk read/write

To revert to a pure disk read/write system, the following paragraph should assist:

The variable "number" keeps track of where you are in the original document, while "count" concerns the output document pointer. Wherever PEEKs and POKES are used in conjunction with these variables they should be exchanged with BGET and BPUT (from Toolkit 2), having first opened the relevant files on disk for read and write, and allocated channel numbers to them. After this, instead of SBYTES (line 3270) you should simply close the files on disk. Further, it will no longer be necessary to reserve areas of memory using the ALLOCATE command (DP's enhanced version of RESPR) so the relevant lines should be deleted.

We now come to the main program loop (lines 2410 to 3260). The following simple, logical structure will be our starting point:

1. Increment the "number" and "count" variables on each turn of the loop (lines 2460 and 2480).
2. Check that an end of processing condition has not been met (line 2470). (Save the file and end if it has.)
3. If nothing has to be done about the relevant character, simply poke it into the new file (line 3060).
4. If extra characters, such as control characters, have to be added into the new document, the "count" variable has to be incremented further to keep track of the length of the new

document (eg line 2700).

5. Keep track of error conditions for future reporting. For this purpose we will use the current line number of the document as a reference point. We will therefore require some sort of array which will record the type of error and the line numbers where the errors occur.

As we must examine each character and take appropriate action, a SELECT variable is an obvious choice, however, you will have noticed from the program listing that after assigning the current character of the source document to the variable n1 (line 2490), there are three sets of SELECT operations (lines 2500 to 2820, 2830 to 3070 and 3080 to 3250). We have not changed the value of n1 between the selects, so why don't we just have one set of selects?

Grouping Selects

Consider the following: there are over 191 characters in the QL set, each of which has to be analysed and subjected to some action, even if it is only to copy it from the source to the object document. There are also various types of operation which have to be performed on these characters. However not all the operations are performed in the same combination, or the same order for those characters to which they apply. It is better, therefore, to group together similar combinations of operations into one set of selects and others into another select.

For example, all the characters must be poked into the object document, all characters except the control codes must cause the tab_posn marker to be incremented, some of the control codes give rise to further control codes being inserted and likewise for certain printable characters.

Rather than have:

```
SELECT ON n1
= 1: do operations a, c, f and g
= 2: do operations c, a, f and g
= 3: do operations a and c
= 4: do operations a, f and g
```

```
1980 REM: status1 occurs when certain special characters
1990 REM: leading to a possible ppnl toggle are found.
2000 REM: status2 occurs when a space is found.
2010 REM: if status2 follows status1 or status2 (i.e.
2020 REM: two spaces) then status3 occurs.
2030 REM: any spaces found after status3 maintain the
2040 REM: status3.
2050 REM: the first printable character found after a
2060 REM: status3 is where a toggle may go - bingo!
2070 REM: This is known as status4
2080 number=number+1
2090 IF number=total: EXIT line_analysis
2100 letter=PEEK(number)
2110 SElect ON letter
2120 =31 TO 32
2130 tab_posn=tab_posn+1
2140 IF status2 OR status1: status3=1:ELSE status2=1
2150 =33 TO 35, 37 TO 38, 41 TO 43, 45 TO 47, 59 TO 64,
123, 125 TO 127, 182 TO 186
2160 tab_posn=tab_posn+1:IF status5=0: status1=1
2200 =10
2210 lno=lno+1:NEXT line_analysis
2220 =0 TO 9,11 TO 26
2230 NEXT rest_of_line
2240 =58
2250 tab_posn=tab_posn+1
2260 IF status5=0: status1=1
2270 IF status3 :status4=1:ELSE status1=0:status2=0
2280 =124
2290 status4=1
2300 =REMAINDER
2310 tab_posn=tab_posn+1
2320 IF status3 :status4=1:ELSE status1=0:status2=0
2330 END SElect
2340 IF status4
2350 status5=status5+1
2360 IF (status5+1)<5:info_array%(lno,(status5+1))=ta
b_posn:ELSE info_array%(lno,4)=tab_posn
2370 status1=0:status2=0:status3=0:status4=0:NEXT res
t_of_line
2380 END IF
2390 END REpeat rest_of_line
2400 END REpeat line_analysis
2410 number=buffer1-1:lno=1:ntab=0
2420 next_tabbed_line=next_tabbed_line+analyse%
2430 uline=0:tab_posn=0:ital=0:bold=0:ppnl=0
2440 elite=0:sscript=0:comp=0:enlarged=0:status7=0
2442 eliterrstat=0:comperrstat=0:italerrstat=0
2444 bolderrstat=0:sscripterrstat=0
2450 REpeat loop
2460 number=number+1
2470 IF number=total:POKE count,10:count= count+1:EXI
T loop
2480 count=count+1
2490 n1=PEEK(number)
2500 SElect ON n1
2510 REM: This select deals with situations requiring
2520 REM: something to be poked or done BEFORE the
2530 REM: character itself is poked into the new
2540 REM: document i.e. proportional insertions,
2550 REM: underline corrections and end of line or
2560 REM: paragraph errors or reports
2570 =31 TO 191
2580 tab_posn=tab_posn+1
2590 IF status7=1:IF n1>32:POKE count,21:count=co
unt+1:uline=1:status7=0
2600 IF lno=next_tabbed_line
2610 IF ppnl AND comp=0
2620 IF status6=0:IF n1>32:POKE count,16:count=co
unt+1:status6=1
2630 IF status6=1:IF tab_posn=next_tabbed_line:POKE count,16
:count=count+1:status6=0:next_tabbed_line=analyse%
2640 ELSE next_tabbed_line=analyse%
2650 END IF
2660 END IF
2670 =10
2680 tab_posn=0:lno=lno+1:status6=0
2690 IF lno>next_tabbed_line: next_tabbed_line=en
alyse%
2700 IF enlarged:enlarged=0:POKE count,7:count=co
unt+1:note_error enlerr,lno-1
2710 IF ssripterrstat=1:note_error ssripterr,lno-1:sscripterrstat=0
2720 IF uline
2730 POKE count,21:count=count+1:uline=0:status7=1
2740 IF PEEK(number+1)=10:status7=0:note_error ul
inerr,lno-1
2750 END IF
2760 IF PEEK(number+1)=10
2770 IF italerrstat:note_error italerr,lno-1:ital
errstat=0
2780 IF eliterrstat:note_error eliterr,lno-1:elit
errstat=0
2790 IF bolderrstat:note_error bolderr,lno-1:bold
errstat=0
2800 IF comperrstat:note_error comperr,lno-1:comp
errstat=0
2810 END IF
2820 END SElect
2830 SElect ON n1
2840 REM: This select deals with characters requiring
2850 REM: poking BEFORE AND/OR AFTER the character
2860 REM: itself.
2870 REM: The remainder is to poke the characters
2880 REM: themselves which were not physically poked by
2890 REM: the previous select.
2900 =142, 143, 145, 146, 149, 152, 155
```


= 5: do operations c, a and g
and so on

It might be better to try the
following:

```
SELECT ON n1
```

= 2.5: do operations c and a
= REMAINDER: do operation a
END SELECT

```
SELECT ON n1
= 1,3: do operation c
END SELECT
```

```
SELECT ON n1
= 1, 2 and 4: do operations f and g
= 5: do operation g
END SELECT
```

It is obvious that we would reduce the length of our program considerably using this method, without any noticeable loss of speed; suppose that the actual statement of "operation a" was 30 characters long: this would not be worth converting into a procedure but, by using the above method, it would only be required to be stated a few times in our program. Also, of course, the more possibilities there are, the greater the saving.

Happy medium

However, if we pursued this logic to the ultimate conclusion by reducing the number of operations to an absolute minimum, we would be losing out by having the maximum number of SELECT algorithms! What we want therefore is the optimum balance between the two. The answer in any case will change if your own requirements are in any way different from mine. Just bear it in mind! As to my particular requirements, I have inserted some REM statements in the listing so that you can see how the SELECTs are distributed.

Our next task is to use this structure to keep track of where our document is as regards printer modes and perform some simple automatic error corrections. I shall take this up in next month's QL World.

```
2910 IF ppnl AND status6=0 AND comp=0: POKE cou
nt,16:count=count+1:POKE count,n1:count=count+1:POKE co
unt,16:ELSE POKE count,n1
2920 =8
2930 IF tab_posn>1:IF tab_posn<=info_array%(lno
,1):note_error comperr,lno
2932 IF comperrstat:comperrstat=0:ELSE comperrstat=1
2940 IF NOT comp
2950 IF elite:POKE count,17:count=count+1:pre
compel=1:ELSE precompel=0
2960 IF ppnl:POKE count,16:count=count+1:pre
compppnl=1:ELSE precompppnl=0
2970 POKE count,8
2980 END IF
2990 IF comp
3000 POKE count,8:count=count+1
3010 IF precompppnl:POKE count,16:count=count+1:precompppnl=0
3020 IF precompel:POKE count,17:count=count+1:precompel=0
3030 count=count-1
3040 END IF
3050 IF comp THEN comp=0:ELSE comp=1
3060 =REMAINDER: POKE count,n1
3070 END SELECT
3080 SELECT ON n1
3090 REM: This section keeps tabs on control codes
3100 =25
3102 IF italerrstat:italerrstat=0:ELSE italerrstat=1
3104 IF ital:ital=0:ELSE ital=1
3110 =21:IF ufile THEN ufile=0:ELSE ufile=1
3120 =2
3122 IF bolderrstat:bolderrstat=0:ELSE bolderrstat=1
3124 IF bold:bold=0:ELSE bold=1
3130 =19
3132 IF sscripterrstat:sscripterrstat=0: ELSE sscripterrstat=1
3134 IF sscript:sscript=0:ELSE sscript=1
3140 =16:IF ppnl THEN ppnl=0:ELSE ppnl=1
3150 =17
3152 IF eliterrstat:eliterrstat=0:ELSE eliterrstat=1
3160 IF elite:elite=0:ELSE elite=1
3170 IF tab_posn>1:IF tab_posn<=info_array%(lno,1):note_error eliterr,lno
3180 =7
3190 IF enlarged
3200 enlarged=0
3210 IF (2*(tab_posn)-enlstart)>80:note_error enlerr,lno:enlstart=0
3220 ELSE
3230 enlarged=1:enlstart=tab_posn
3240 END IF
3250 END SELECT
3260 END REPEAT loop
3270 SBYTES destfile$,buffer2,(count-buffer2)
3280 ADVISE
3290 DEALLOCATE buffer1:DEALLOCATE buffer2
3300 END REPEAT main_loop
3310 DEFINE FUNCTION analyse%
3320 IF next_tabbed_line=0:RETURN 0
3330 store=lno
3340 REPEAT POSSIB
3350 store=store+1:IF store>lines:RETURN 0
3360 FOR better=4 TO 2 STEP -1
3370 tab=info_array%(store,better)
3380 IF tab>2
3390 SELECT ON tab
3400 =3 TO 15:offset=40
3410 =16 TO 20:offset=30
3420 =21 TO 25:offset=25
3430 =26 TO 34:offset=20
3440 =35 TO 45:offset=15
3450 =REMAINDER :offset=4
3460 END SELECT
3470 success=0
3480 IF offset<store:backstore=store-offset: ELSE backstore=1:
3490 IF lines>=(store+offset):frontstore=store+offset:ELSE frontstore = lines
3500 FOR round=backstore TO frontstore
3510 IF info_array%(round,1) = tab OR info_array%(round,2) = tab OR info_array%(round,3)=tab OR info_array%(round,4)=tab:success=success+1
3520 IF success>2:nltab=tab:RETURN store
3530 NEXT round
3540 END IF
3550 NEXT better
3560 END REPEAT POSSIB
3570 END FUNCTION analyse%
3580 REMARK -
3590 DEFINE PROCEDURE wait
3600 REPEAT poll
3610 key$=INKEY$(#4,-1):IF key$<>"":PAUSE 20:EXIT poll
3620 END REPEAT poll
3630 END DEFINE wait
3640 DEFINE PROCEDURE ADVISE
3650 PRINT #4, " Time taken: ";(DATE-n); " seconds
3660 FOR n=1 TO 7:tell_error(n)
3670 PRINT #4, " Proportional: ";:IF ppnl:PRINT #4, "ON. ":ELSE PRINT #4, "OFF. "
3680 wait:CLS #4
3690 END DEFINE ADVISE
3700 DEFINE FUNCTION YEA_OR_NAY
3710 PRINT #4, " (Y/N) ? ";:CURSOR_ON #4
3720 REPEAT poll
3730 key$=INKEY$(#4,-1):IF key$=="y" OR key$=="n"
3740 PRINT #4, key$:CURSOR_OFF #4
```

```

3750 RETURN Key$=="y"
3760 END IF
3770 END REPEAT poll
3780 END DEFine YEA_OR_NAY
3790 DEFine PROCedure note_error (a,b)
3800 errcount_array(a)=errcount_array(a)+1
3810 IF errcount_array(a)>10:RETURN
3820 error_array(a,errcount_array(a))=b
3830 END DEFine note_error
03840 DEFine PROCedure tell_error(a)
3850 SElect on n
3860 =1:tmp$="Elite":c=elite
3870 =2:tmp$="Compressed":c=comp
3880 =3:tmp$="Enlarged":c=enlarged
3890 =4:tmp$="Superscript":c=sscript
3900 =5:tmp$="Bold":c=bold
3910 =6:tmp$="Italics":c=ital
3920 =7:tmp$="Underline":c=uline
3930 END SElect
3940 PRINT #4, "      " & tmp$ & ": ";
3950 IF c=1:PRINT #4,"ON. ":ELSE PRINT #4,"OFF. "
3960 IF errcount_array(a)>10:errcount_array(a)=10
3970 IF errcount_array(a)
3980 PRINT #4, "      ";er$;
3990 FOR tally=1 TO errcount_array(a):PRINT
#4, error_array(a,tally); " ";
4000 END IF
4010 wait:CLS #4
4020 END DEFine tell_error

```

Perfect Paragraphs and Shining Characters

QUBBESoft P/D

3.5inch
Disks
50p
Copying
Fee

Public Domain, Shareware & 2nd User Software
New & 2nd User Hardware
For The

Sinclair Q.L.

MDV's
30p
Copying
Fee

NEW Q.L. Trump Card

When the Q.L. Trump Card is plugged into the Q.L. an extra 768K of RAM is added to the Q.L.'s own memory to give an amazing 896K of RAM in total. Not only do you get an increase in memory but included in the package is the famous Toolkit II from QJump, which gives you over 100 new Commands & Functions. This is not all, also included is a Disk Interface which enables the User to connect most types of Disk Drives to the Q.L. There is an A4 size Manual which explains all the new Commands & Functions available. All this for the amazingly low price of:-

£95.00p (Plus Postage & Packing)

NEW Q.L. ExpanderRAM

When the Q.L. ExpanderRAM (512K) is plugged into the Q.L. an extra 512K of RAM is added to the Q.L.'s own memory to give a total of 640K of RAM. The Q.L. ExpanderRAM has a through port enabling the User to connect other Peripherals such as a Disk Interface, EPROM Programmer, EPROM Board, etc. etc. 2 Versions of the ExpanderRAM are available. 1 being fully populated with RAM chips i.e. 512K, the other having no RAM chips i.e. OK. For the low price of:-

ExpanderRAM (512K) £45.00p ExpanderRAM (OK) £20.00p (Plus P & P)

NEW MicroEMACS V3.11 Text Editor

2 Disks

Improvements over the QL Version 3.9p include:- More speed (Compiled with the latest C68), Pop-up buffers (e.g. for buffer, screen & files lists), Improved support for external programs (piping & filtering), Multiple marks (currently 10), Built-in indent and undent commands, Impressive help system, and much much more too many to list here. All for the low price of.

£2.00p (Pre-Copied Disks) £1.00p (User Supplies Disks) Plus P&P.

Postage & Packing

Trump Card & ExpanderRAM £3.00p (UK) £5.00p (Europe) £7.50 (ROW)
MicroEMACS V3.11 30p (UK) 70p (Euro) £1.00 (ROW)

Address

38, Brunwin Road, Rayne, Braintree, Essex. CM7 5BL.
Tel No. (0376) 347852.

Payment

Cheques/Postal Orders payable to:- K.Dunnell
Developments in the Pipeline

Software:- Postler Environment Utilities,
Hardware:- EPROM Programmer.

Catalogue

Stamped Address Envelope (A5 Size) or I.R.C.

C.G.H. SERVICES

CWM GWEN HALL, PENCADER, DYFED, CYMRU, SA39 9HA
PUBLIC DOMAIN AND SHAREWARE LIBRARY
ALL DISKS £2.00 EACH INCLUSIVE OF MEDIA AND U.K. P&P
PLEASE ADD 10% FOR POSTAGE TO EUROPE, 20% FOR REST OF WORLD
PAYMENT IN CASH, CHEQUES, POSTAL ORDERS ETC IN STERLING.
(WE CAN ALSO ACCEPT PAYMENT IN CASH FROM GERMANY AND CANADA)

ADVENTURE GAMES DISK 1
ADVENTURE GAMES SOURCE CODE DISK 1:
ADVENTURE SOLUTION DISKS 1-3
ADVENTURE SOLUTIONS DISK 4: QL specific
ADVENTURE UTILITIES DISK 1
AUSTRALIAN P.D. DISKS 1+2
BOOT PROGRAMS DISK 1: a small selection.
COMMUNICATIONS DISK 1
COMMUNICATIONS DISK 2: includes QL Keenit.
CONNECTIONS DISK 1
DEVICE UTILITIES DISK 1
DEVICE UTILITIES DISK 2: includes archiving
utils
DILWYN JONES DISK 1: over 20 progs
DISK CATALOG DISK 1: Diskidy program.
EDITORS DISK 2: QED and MicroEmacs runtime.
EDITORS DISK 3: MicroEMACS source code.
EDUCATIONAL PROGRAMS DISK 1
EMMANUEL VERBEEK DISK 1
EMULATORS DISK 1: ZX81 and Spectrum
ENGLISH DICTIONARY DISKS 1 AND 2
ESOTERICA DISK 1
POINTS DISK 1: A few text 87 fonts - more wanted.
FRACTALS DISK 1: large number of progs
FRACTALS DISK 2: Carl Cronia's Mandelbrot prog
FRACTALS DISK 3: Rainer Kowalick's prog
FRACTALS DISK 4: sample screens from Q.Ractal
FRACTALS DISK 5: Lyapunov Space generator
GAMES DISK 1: includes Starburst, Cavern Fantasy.
GAMES DISK 2: includes many arcade type games.
GAMES DISK 3: includes QL War.
GAMES DISK 4: includes Odyssey (not Minerva)
GERMAN - ENGLISH DICTIONARY DISK
GRAPHIX DEMOS DISK 1: Too many to list!
GRAPHIX DEMO DISK 2: "Movieclon"
GRAPHIX DEMO DISK 3: "Elite" Demos
GRAPHIX DEMO DISK 4: Double 3-D Wave
GRAPHIX DEMO DISK 5: Wirtworld (512K+)
GRAPHIX DEMO DISK 6: "Turning Head"
(see Minerva)
GRAPHIX DEMO DISK 7: Mark J. Swift's
"Turning Head" done for Minerva.
GRAPHIX SCREENS (GIF FORMAT) DISKS
1,2,3: Misc screens with GIF viewer prog.
GRAPHIX SCREENS (PCX FORMAT) DISK 1:
Comes with screens and viewer prog. (Not Minerva)
GRAPHIX SCREENS (QL) DISK 1: Misc screens
GRAPHIX SCREENS (QL) DISK 2: clip art
GRAPHIX SCREENS (QL) DISK 3: clip art
GRAPHIX SCREENS (SPECTRUM) DISK 1:
GRAPHIX SCREENS (ST) DISKS 1,2,3,4,5,6,7,8:
GRAPHIX/SCREEN UTILITIES DISK 1:
HAM RADIO DISK 1: includes PC conversions.

HARDWARE INFO DISK 1: Quest & SuperQBoard
INDEXES DISK 1: QL World and QUANTA
JIMMIE ROBB DISK 1: printer progs
MALCOLM SMITH DISK 1: 5 Superbasic progs
MATHS AND CALENDAR DISK 1:
MONEY DISK 1: need extra memory and TK2.
OLIVER PINK DISK 1: needs Pointer environment
PRINTER UTILITIES DISK 1
PROGRAMMING DISK 1: includes QL PROLOG.
PROGRAMMING DISK 9: includes QLPORTH
PROGRAMMING DISK 18: C Tutorial
PROGRAMMING DISKS 19 - 21: C68 For QDOS
Runtimes and docs. (By Dave Walker/Release 3.02)
PROGRAMMING DISKS 22- 26: C68 For QDOS,
Source Code. (Release 3.02)
PROGRAMMING LANGUAGES DISK 27:
C-PORT Support Library.
QITALY DISKS 5 - 11: mostly in Italian.
QFAC UTILITIES DISK 1: QFAC program
RALF BIEDERMANN DISKS 1+2
RAY DENT DISK 1: 4 utility progs.
RECREATIONAL MATHS DISK 1: Life, etc
RICH MELLOR DISK 1: 4 utility progs.
STEPHEN BEDFORD DISK 1: 7 utility progs.
SUPERBASIC UTILITIES DISK 1: Loads of progs
TEXT DISKS 1 - 7: The Bible - (Old Testament)
TEXT DISKS 8 - 9: Computer Jargon Text Files.
TEXT DISKS 10 - 11: German-English Word
Translator (and vice versa)
TEXT DISK 12: Olympic Winners (up to 1988!)
TEXT DISK 13: Misc files
TEXT DISK 14: 400+ Business Letters from U.S.A.
TEXT DISKS 15 - 16: The Bible (New Testament)
TEXT DISKS 17 - 21: 200,000 English words.
TEXT DISKS 22 - 24: CIA's World Fact Book.
THOR P.D. DISK 1: THOR info files and progs.
T.V. MOVIE DISKS 1 - 3: Digitized TV animations.
TYPING TUTOR DISK 1: Artip prog.

PLEASE SEND A LARGE STAMPED
SELF-ADDRESSED ENVELOPE TO THE
ABOVE ADDRESS TO RECEIVE DETAILS
OF OUR COMMERCIAL QL PROGRAMS,
QL MAGAZINES AND D-I-Y TOOLKIT.

WE CAN ALSO SUPPLY A WIDE RANGE OF
COMMODORE, ATARI AND GOLDSTAR
COMPUTERS, PRINTERS, EXTERNAL DISK
DRIVES AND OTHER PERIPHERALS.
GAMES CONSOLES AND SOFTWARE FOR
THE ST, AMIGA, PC AND GAMES
CONSOLES. (U.K. CUSTOMERS ONLY)
PLEASE PHONE 0559-384574 FOR DETAILS



Dilwyn Jones Computing

**41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT U.K.**

Tel: Bangor (0248) 354023

GRAPHICS

THE PAINTER V4.04 £25.00
(F 512k) 100% machine code pointer driven art and graphics program, by PROGS of Belgium. Can be mouse controlled. Patterns, fills, size changes, screen dumps, all sorts of facilities.
THE CLIPART £12.00
(F 128k) 3 disks packed full of screen pictures for use in desktop publishing and graphics programs. Ideal for use with The Painter or Image Processor or Page Designer, for example!
QRACTAL £20.00
(F 512k) Machine code fractals program, able to display Julia or Mandelbrot sets.
QRACTAL SCREENS £5.00
(F 128k) Sample screen pictures made with QRACTAL.
IMAGE PROCESSOR 2 £15.00
(F 512k) Picture editing and creation system, image enhancement, mode conversion, touch up pictures from other sources. Can be used to tidy up screens from digitisers, for example, or just as an art and graphics program.
PD2 CLIPART £10.00
(F 128k) Clipart from Page Designer 2, supplied in mode 4 screens format for use with any graphics program.
SCREEN SNATCHER £10.00
(F M 128k) Grab screen displays from within other programs for use in your own programs!
TEXT 'N' GRAPHIX £20.00
(F 256k) Print text files containing screen dumps, mix text and graphics in same document.
TRANS24
(F M 128k) Translate 9 pin graphics output from most programs without a 24 pin printer driver to 24 pin.

TEXT

QIYP 2 £29.95
(F 512k) Typing and spelling checker, which uses pointer environment, 40,000 word English dictionary plus French and German dictionaries. Dictionary editor included to add your own words. SuperBASIC interface allows you to write your own BASIC programs which use QIYP.
BIBLE TEXT DISKS, PLAIN TEXT FORMAT £20.00
BIBLE TEXT DISKS, QUILL FILE FORMAT £20.00
(F M 384k) Text of the King James Bible on disks.
SPELLBOUND £30.00
(F M 384k) Spelling checker which checks your typing as you type 30,000 word dictionary, expandable.
SPELLBOUND SPECIAL EDITION £50.00
(F 512k) Improved version with larger 50,000 word dictionary (expandable). Check spelling as you type or check existing text files.
UPGRADE TO SPECIAL EDITION £30.00
Please return master disk/cartridge when upgrading.
QUICK POSTERS £10.00
(F M 128k) Text poster maker for use with Star LC, XL and XB printers. Current version works in pointer environment, upgrade from old version £3.00
ROB ROY BARGAIN PACK £10.00
(F M 128k) See review in QL World August 1991

DATABASES

ADDRESS BOOK & LABEL PRINTER £15.00
(F M R 384k) Store names and address and print them out on a variety of label sizes, print telephone lists, etc.
DATA DESIGN £50.00
(F 512k) A database which can be programmed from BASIC or machine code. Runs in pointer environment which is supplied with the program. Fast and versatile.
FLASHBACK £25.00
(F M 256k) Fast database written in machine code, not programmable.
FLASHBACK SPECIAL EDITION £40.00
(F 256k) Enhanced version of Flashback, with Report Generator, etc. NB Please advise if to be used on a Gold Card, since Reporter Generator requires modification.
QL GENEALOGIST SECOND EDITION £30.00
(F 384k) Family trees and family history program, one of our best selling programs!
UPGRADE TO SECOND EDITION £12.00
Please return master disk with upgrade order.
BUDGET 128K GENEALOGIST £12.00
(F M 128k) A cut down version version for unexpanded machines, still superb value for money.
DISK INDEXER £12.00
(F M 128k) Make a database of the contents of your disks, so that you can search for files, list them, and generally tidy up your disks and cartridges!
DBEASY £15.00
(F 512k) Database front end for Archive and a suite of programs called Chaos Busters!
DBPROGS £15.00
(F 512k) A collection of Archive utilities and text files to help you to learn to program Archive.

D.T.P.

PAGE DESIGNER 2 PLUS £40.00
(F 512k) QL Desktop publishing program, mix text and graphics, print pages, use graphics from other programs. Prints on most printers. Ask for information. Sadly, it has been badly delayed, so you should check with DJC first to see if available yet before ordering.
UPGRADE TO PD2 PLUS £20.00
Send proof of purchase of old Page Designer 2 with upgrade order. See note above re. delays.

QL HARDWARE

MINI PROCESS CONTROLLER £59.95
Relay switched outputs, controlled via QL serial ports.
SOFTWARE TOOLKIT FOR MPC £9.95
(F 128k) BASIC extensions to simplify writing programs to control the MPC.
NETWORK PROVER £3.50
A small box which plugs between networked QLs to give a visual indication of transmission over the network.
QPOWER REGULATOR £24.95

OTHER QL SOFTWARE

QPAC1 £19.95
(F 512k) Pointer environment utilities, including calculator, calendar, alarm, typewriter, clock and system monitor utility. Pointer environment included. Quality QL software by Tony Tebby.
QPAC2 £39.95
(F 512k) If you want to make full use of multitasking on your QL, this package is a must. Files maintenance menu, hotkeys "things", channels, pick, etc. Extensive documentation and a tutorial is supplied. Another Tony Tebby package.
QTOP £29.50
(F 256k TK2) User front end for QDOS by Cowo Electronic of Switzerland. Desktop system, file handling, job control, program startup menus, with 5 desktop accessory programs.
PRINTERMASTER £20.00
(F M 128k) Printer control utility - select fonts, set margins, page length and much more. Printermaster puts you in control of your printer. Not a graphic program, not a printer driver creator, just a database of printer functions with the ability to send control codes to the printer quickly and easily without having to reach for your manual. Files for several makes of printers supplied, or create your own or modify those supplied quickly and easily. In constant use at DJC, that's what I think of it!
HOME BUDGET £20.00
(F M 128k) Domestic bills and accounts program.
REMIND-ME £12.00
(F M 128k) Dates reminder program. Quick and easy to use.
REMIND-ME PLUS £20.00
(F M 128k) Enhanced to allow more dates to be scheduled, longer event descriptions.
SCREEN ECONOMISER £10.00
(F M 128k) Turns off the QL display after a set number of minutes during which QL has not been used to protect the screen.
SLOWGOLD £5.00
(F 128k) Slowdown routine and control panel for software which runs too fast on a Gold Card or indeed on an QL system.
TASKMASTER £25.00
(F M 384k) Task switching utility. Enables you to conveniently switch between several programs in memory. Calculator, notepad and file handling utility included. Ideal if you want a simple to use system to let you have several programs in the computer at once and to switch between them at the press of a couple of keys. Can even save and restore the display of each program automatically.
DISK LABELLER £10.00
(F 256k) Prints neat labels for your floppy disks, listing the filenames from the disk in columns in small print on the label. Essential starting point to tidying up your disk collection. We have sold a great deal of copies of this useful program - have you got yours yet?
THE CAT £5.00
(F M 128k) List filenames from a disk or cartridge in columns to the screen or printer. The list can be sorted if required. Useful and convenient utility. Installs as a BASIC extension then just type CAT for a list of files.

THE SMALL PRINT! POSTAGE AND PACKING CHARGES: Software is sent post-free to UK addresses. To other countries, please add £1.00 per program for postage and packing.
PRICES: All prices are shown in UK Pounds Sterling. **PAYMENT:** We can accept payment by cheque (in UK Pounds Sterling currency only, please, drawn on UK branch of a bank or building society), Eurocheque, Postal Order, or by these credit cards: VISA, ACCESS, MASTERCARD, EUROCARD and by CONNECT card. Please state the card type, number, expiry date, your address and sign orders sent by post. We can also accept orders paid by credit card by telephone. There is an answering machine for when I cannot answer in person, so that I can call you back later. Goods remain the property of DJC until paid for in full. **PLEASE STATE IF YOU REQUIRE 3.5 OR 5.25 INCH DISKS OR ON MICRODRIVE CARTRIDGE (ADD £1 PER PROGRAM ON MICRODRIVE CARTRIDGE TO COVER HIGHER COSTS OF SUPPLYING SOFTWARE ON MICRODRIVE).**